

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

INTEGRATION OF AN APPLE II PLUS COMPUTER
INTO AN
EXISTING DUAL AXIS SUN TRACKER SYSTEM

by

Roger J. Morais

June 1984

Thesis Advisor:

H. Titus

Approved for public release; distribution unlimited.

T223005

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|--|-----------------------|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) Integration of an Apple II plus Computer into an Existing Dual Axis Sun Tracker System | | 5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1984 |
| 7. AUTHOR(s) Roger J. Morais | | 6. PERFORMING ORG. REPORT NUMBER |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943 | | 8. CONTRACT OR GRANT NUMBER(s) |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 12. REPORT DATE June 1984 |
| | | 13. NUMBER OF PAGES 102 |
| | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Sun Tracker System Solar Energy Apple II Plus Computer | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis describes the integration of an Apple II plus computer into an existing sun tracking system. The Apple Computer replaced an Intel 80/10A single board computer as the system controller. Software development and hardwiring were necessary to successfully integrate the new computer into the system. With the new computer installed, user interaction with the tracking system became possible. Additionally it was | | |

#20 - ABSTRACT - (CONTINUED)

possible to replace hard to interpret assembly language code with higher level Basic code as the system controlling software.

Approved for public release; distribution unlimited.

Integration of an Apple II plus Computer
into an
Existing Dual Axis Sun Tracker System

by

Roger J. Morais
Lieutenant, United States Navy
B.S., University of Washington, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1984

ABSTRACT

This thesis describes the integration of an Apple II plus computer into an existing sun tracking system. The Apple Computer replaced an Intel 80/10A single board computer as the system controller. Software development and hardwiring were necessary to successfully integrate the new computer into the system. With the new computer installed, user interaction with the tracking system became possible. Additionally it was possible to replace hard to interpret assembly language code with higher level Basic code as the system controlling software.

TABLE OF CONTENTS

| | | |
|------|---|-----|
| I. | INTRODUCTION ----- | 7 |
| II. | SYSTEM DESIGN ----- | 13 |
| | A. SYSTEM COMPONENT DESCRIPTION ----- | 13 |
| | B. SYSTEM STRUCTURE ----- | 18 |
| | C. SYSTEM OPERATION ----- | 24 |
| | D. COMPUTER INTEGRATION ----- | 36 |
| III. | SOFTWARE DESIGN ----- | 40 |
| | A. INITIAL PLANNING ----- | 40 |
| | B. DRIVE ROUTINE DESIGN ----- | 40 |
| | C. DESIGN OF SIGNAL INPUT ROUTINES ----- | 50 |
| | D. DETERMINATION OF SENSOR VALUES ----- | 54 |
| | E. FLASHLIGHT FOLLOW ROUTINE ----- | 55 |
| | F. SUN TRACKING PROGRAM ----- | 57 |
| IV. | CONCLUSIONS AND RECOMMENDATIONS ----- | 62 |
| | APPENDIX A: FLASHLIGHT FOLLOW ROUTINE ----- | 64 |
| | APPENDIX B: SUN TRACKING PROGRAM ----- | 74 |
| | LIST OF REFERENCES ----- | 101 |
| | INITIAL DISTRIBUTION LIST ----- | 102 |

LIST OF FIGURES

| | |
|---|----|
| 1. A Flatplate Collector, and the Sun's Track for a Typical One Day Period ----- | 9 |
| 2. A Point Concentrating Collector ----- | 11 |
| 3. Dual Axis Sun Tracker (Front View) ----- | 14 |
| 4. Dual Axis Sun Tracker (Side View) ----- | 15 |
| 5. Solar Tracker System Block Diagram ----- | 17 |
| 6. Plug Wiring Diagram for Top of Amplification/ Isolation Box ----- | 19 |
| 7. Schematic Diagram of Circuit Board Inside Amplification/Isolation Box ----- | 22 |
| 8. Power Transistor Layout Inside Amplification/ Isolation Box ----- | 25 |
| 9. Intel 80/10A Single Board Computer Connections ---- | 26 |
| 10. Schematic Diagram of Typical Stepping Motor ----- | 29 |
| 11. Four Step Drive Sequence ----- | 32 |
| 12. Eight Step Drive Sequence ----- | 33 |
| 13. Sensor Signal Amplification Circuit ----- | 35 |
| 14. Analog to Digital Card with Connecting Jacks ----- | 37 |
| 15. A488 Drive Routine ----- | 42 |
| 16. 25 Second Drive Routine ----- | 45 |
| 17. 12 Second Drive Routine ----- | 46 |
| 18. 4.8 Second Eight Step Drive Loop ----- | 48 |
| 19. 2.6 Second Four Step Drive Loop ----- | 49 |
| 20. Drive Routine Varying Single Step Element ----- | 51 |
| 21. Display Routine ----- | 53 |

I. INTRODUCTION

As the finite conventional energy sources of the world are depleted, and their costs increase, alternative energy sources are becoming more and more attractive. Although in many instances these alternative sources cannot compete economically with gas, oil, and other conventional sources, it is just a matter of time until alternative and conventional sources will be competitive. Already many homes have solar collectors and other devices designed to decrease the consumption of gas, oil, and electricity. Prototype solar power plants are appearing in various parts of the world, and geothermally produced power is now a viable option in areas where this natural resource is available. In short, our energy sources are shifting, and by necessity will continue to shift, from the non-renewable to the renewable.

Of the renewable energy sources available, solar energy is by far the most abundant. On the average, 1.7×10^{14} kilowatts of solar power continually reach the Earth. If all of this power could be captured and used, it would amount to approximately 5000 times the rate at which power is currently consumed by the entire Earth's population [Ref. 1]. It should be clear that the development of techniques for the conversion of this resource into usable forms of energy is essential to our future as an industrial nation.

One method that is available for converting solar energy into a usable form involves the use of solar thermal collectors. This type of collector captures the sun's radiated energy and transfers the resultant thermal energy to a fluid, which in turn transfers its thermal energy to the user, or places it in a storage facility [Ref. 1].

Solar collectors can be divided into broad categories: the nonconcentrating flat plate collectors, and the concentrating collectors. Flat plate collectors are for the most part stationary, and do not concentrate the energy received by them. In the northern hemisphere flat plate collectors are normally set facing south, and are tilted back to an angle equal to the local latitude plus 10 degrees as shown in Figure 1. In this position the flat plate collector is able to efficiently convert radiated energy into thermal energy for a large range of incidence angles. Point concentrating collectors on the other hand require an incidence angle close to zero to effectively focus the captured radiated energy onto a small area, or a receiver vessel. This means that, in order to maintain a usable incidence angle, either the reflector or its associated receiver must move in response to the sun's changing celestial position. The greater the concentration factor of the particular collector, the more critical the incidence angle of the sun's rays becomes.

The decision to use one collector type over another depends to a large extent upon its intended use. For low

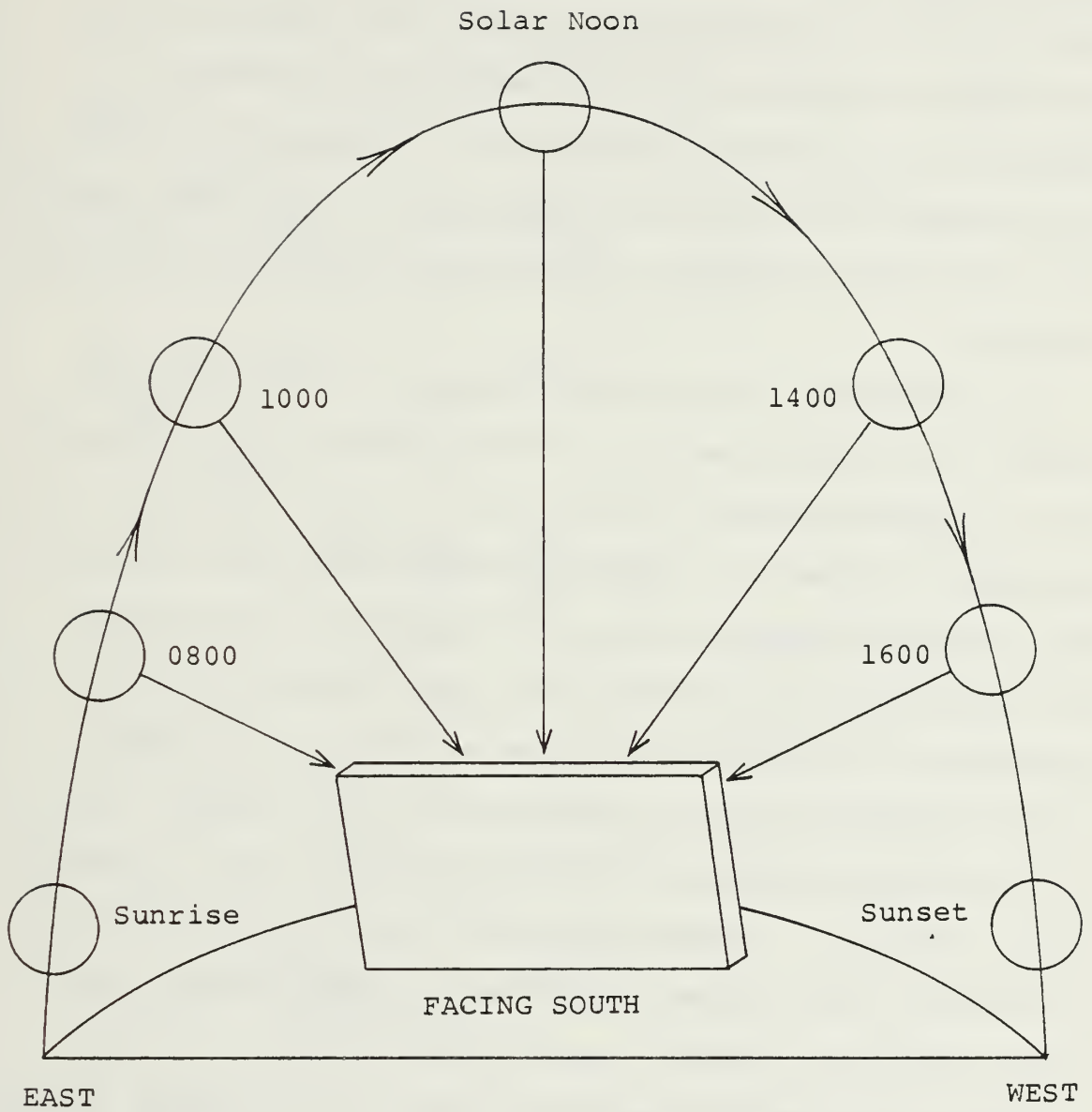
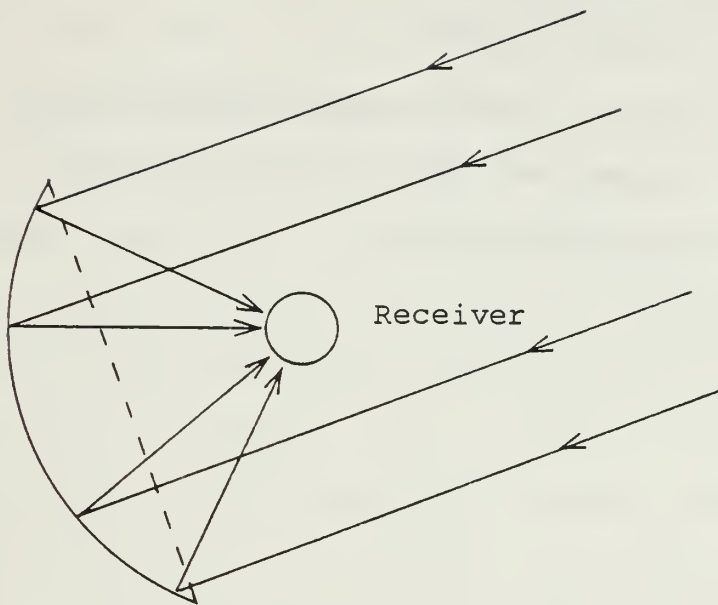


Figure 1. A Flat Plate Collector and the Sun's Track for a Typical One Day Period

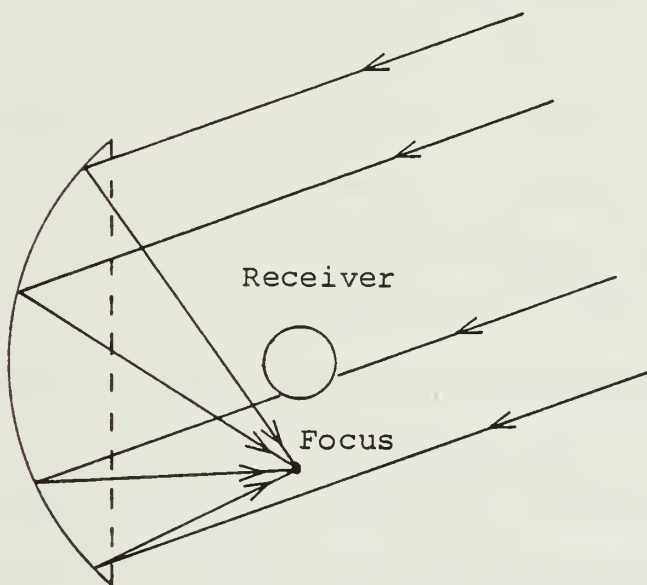
temperature applications the fixed plate collector would normally be the correct choice, but for higher temperature applications the concentrating collector should be used. Power plant applications, for example, use concentrating collectors to produce the high temperatures necessary to generate turbogenerator drive steam. Household hot water and swimming pool heating systems, on the other hand, would use flat plate collectors for both economical and practical reasons.

As mentioned, a point concentrating collector's efficiency is a direct function of how accurately the collector can be pointed directly at the sun. A point focusing type collector is usually a mirror surfaced parabolic reflector. This reflector captures the solar radiation that strikes it, and reflects the energy to a point called the focus. A receiver is placed at the focus and is heated to very high temperatures by the concentrated solar energy. Liquid in the receiver absorbs the heat and can be used to perform work, Figure 2A. If the reflector is not pointed directly at the sun, the incidence angle is not zero and the focus shifts from the receiver, as in Figure 2B.

The mechanism that keeps the point concentrating collector pointed at the sun is the main topic of this research. An Apple II plus micro computer is used to control a Dual Axis Sun Tracking system. The system used already existed, but was controlled by an Intel 80/10A Single Board Computer [Ref. 2].



(a) Incidence Angle 0°



(b) Incidence Angle 20°

Figure 2. A Point Concentrating Collector

Additionally the controlling software was written in assembly language, and downloaded onto EPROM'S. The 80/10A was replaced by an Apple II plus computer to (1) allow for user interaction with the sun tracking system and (2) to have the controlling software written in the easier to understand Basic programming language.

II. SYSTEM DESIGN

A. SYSTEM COMPONENT DESCRIPTION

The solar tracking system consists of three main components. The first is a mechanism known as a Dual Axis Solar Tracking Device. This device consists of two stepping motors for horizontal and vertical drive power, four photo diode sensors used for sun location and tracking, and four limit switches that ensure the tracker drive axis are limited in the amount they can rotate, thus preventing system wrap around, and providing start position information. Figures 3 and 4 show front and side views of the solar tracking component. Figure 3 shows the location of the two stepping drive motors, and the layout of the photo diode sensors (ES,WS,US,DS). As shown the horizontal and vertical stepping motors drive the sensor platform and sensor arm through a series of gears. The horizontal motor causes the platform to rotate through a maximum 180 degree arc in either direction, while the vertical motor causes the sensor arm to move through a maximum 90 degree arc between the up and down positions. Figure 4 shows the sensor arm, the limit switches (EL,WL,UL,DL), and the signals into and out of the tracker. The signals consist of four electrical current values that are proportional to the intensity of the light falling on each sensor, four ON (+5V) or OFF (-5V) signals from the

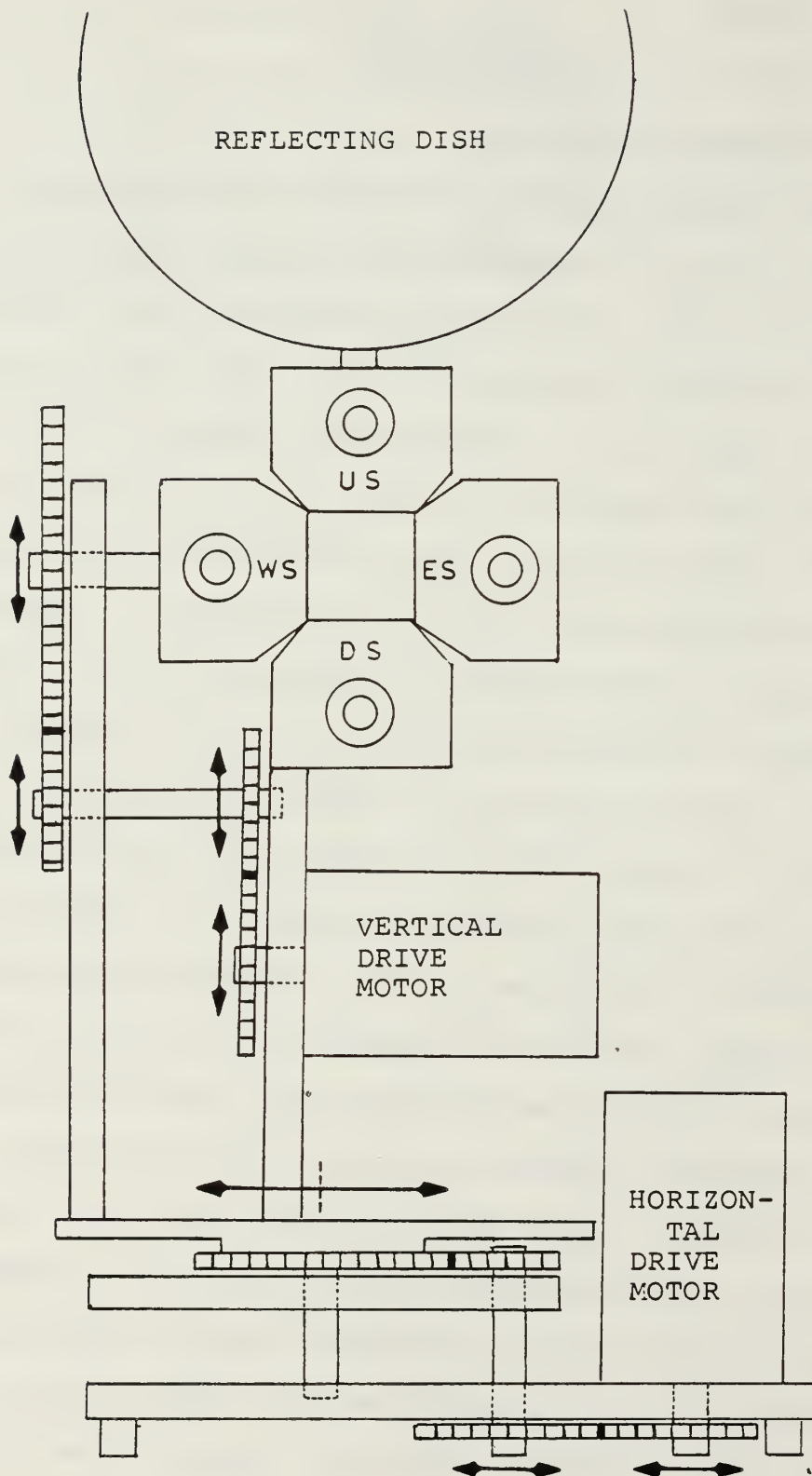


Figure 3. Dual Axis Sun Tracker (Front View)

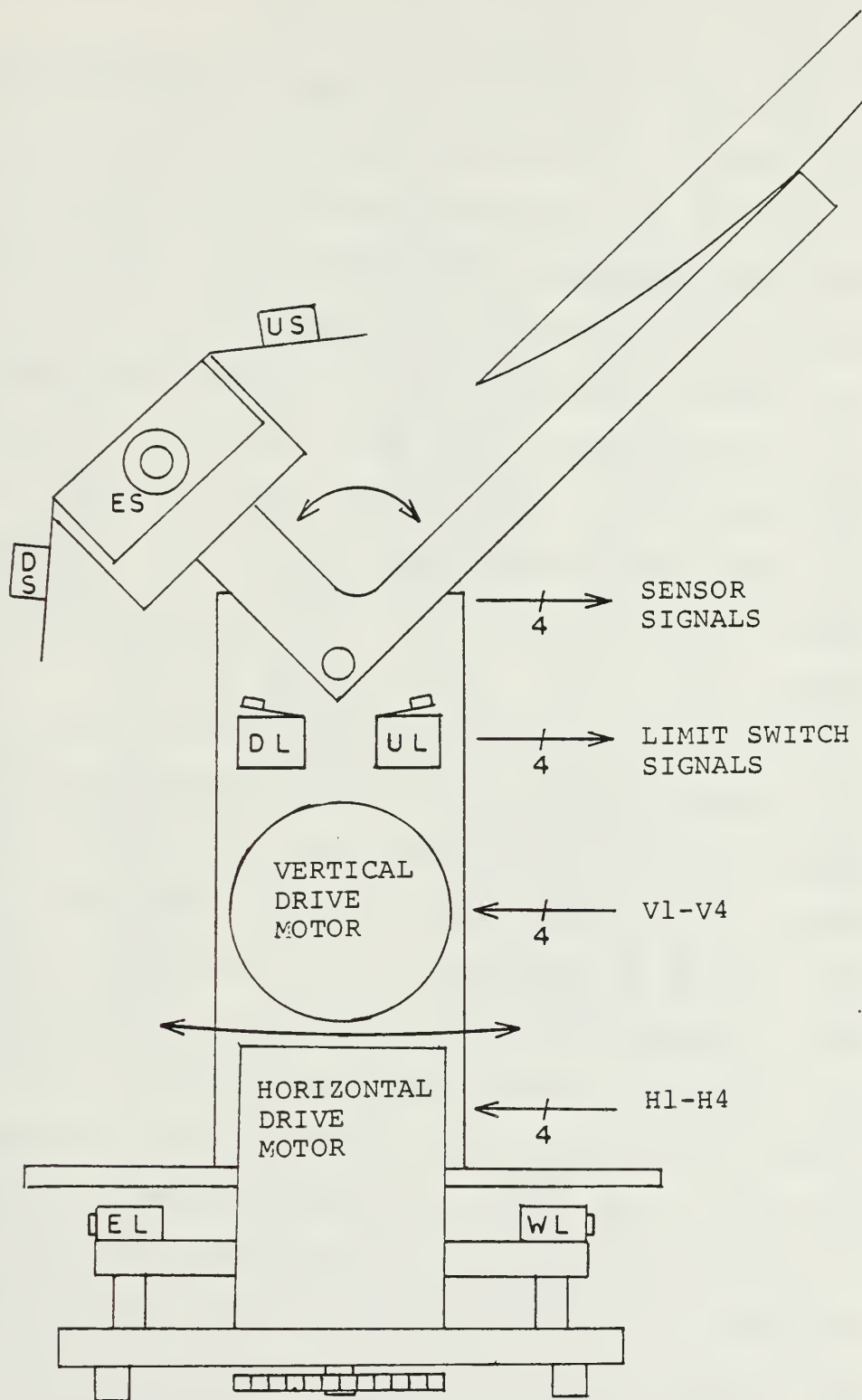


Figure 4. Dual Axis Sun Tracker (Side View)

limit switches, and drive signals from the controlling computer to the stepping drive motors.

The next component of the system is called the Amplification/Isolation box, hereafter called the A/I box. This box contains the circuitry necessary to interface the controlling computer to the sun tracker. Motor drive commands arriving from the computer are amplified and sent on to the drive motors, and sensor signals are amplified before being sent to the computer. Also associated with the amplification circuitry are isolation circuits to protect the computer from short circuits and power surges. The limit switch signals also go through the A/I box on their way to the computer, but are not altered in the process.

The third main component is the Apple II plus computer. This component with its associated software takes in the sensor data and provides the proper drive motor control signals to locate and track the sun. Limit switch values are also checked to determine if the tracker has reached the limit of a particular drive direction. As mentioned the Apple replaced the Intel computer so that the system would be user interactive, and the software would be easier to interpret in the higher level Basic language. Figure 5 is a system block diagram which shows the signal flow between components, as well as the internal flow of the computer. The analog to digital card and controlling software within the computer will be discussed in detail later.

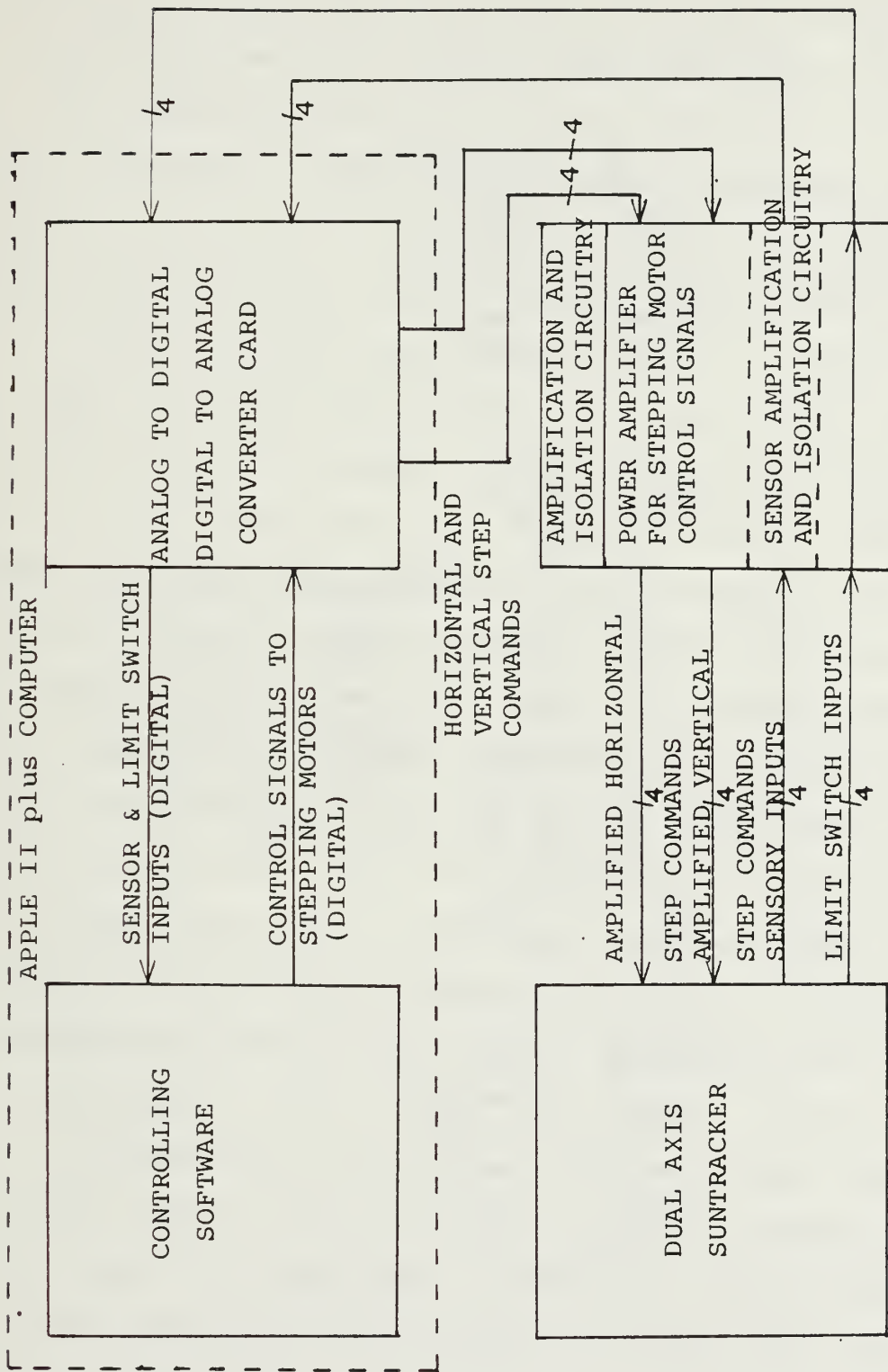


Figure 5. Solar Tracker System Block Diagram

B. SYSTEM STRUCTURE

While some system operation and hardware descriptions were available in [Ref. 2], they were insufficient to provide the detailed system knowledge necessary to integrate a new computer into the system. It was therefore necessary to begin the integration process by performing a thorough system trace out.

The A/I box was the initial and main focus of this trace out since all system signals enter and exit this component. First, all wires entering and leaving the box were traced. Figure 6 and Table 1 describe the results of this effort. Jacks one through three and single wire connections A through V are illustrated with their associated inputs and outputs, these inputs and outputs are further described in Table 1. It should be noted that the micro-processor referenced in the signal descriptions is the original Intel 80/10A and not the Apple. These signals are provided since the Intel hook up was left intact so it can still be used if desired. The Apple is connected inside the A/I box via a bus bar. Once the input and output connections were determined, the A/I boxes internal electronics could be traced. Figure 7 and Table 2 show the layout of the circuit board inside the A/I box, and describe the signals entering and leaving it. Note that the four LM747 OPAMP chips are all connected up identically, but only the east sensor (ES) hook up is drawn to avoid cluttering the

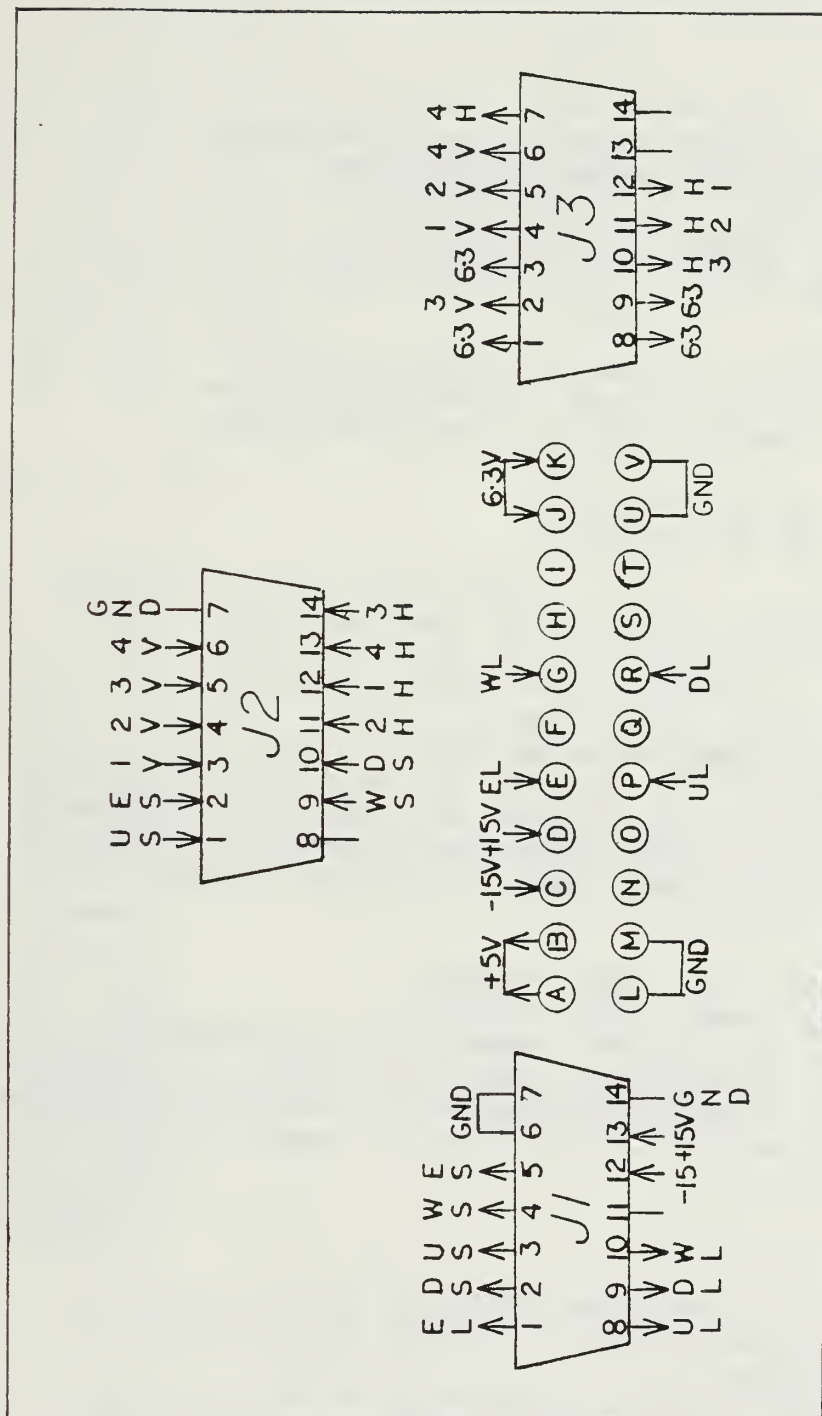


Figure 6. Plug Wiring Diagram for Top of Amplification/Isolation Box

TABLE 1

SIGNAL DESCRIPTION FOR TOP OF AMPLIFICATION/ISOLATION BOX

JACK J1

- (1) EAST LIMIT SWITCH SIGNAL TO MICROPROCESSOR
- (2) DOWN SENSOR SIGNAL TO MICROPROCESSOR
- (3) UP SENSOR SIGNAL TO MICROPROCESSOR
- (4) WEST SENSOR SIGNAL TO MICROPROCESSOR
- (5) EAST SENSOR SIGNAL TO MICROPROCESSOR
- (6) GROUND
- (7) GROUND
- (8) UP LIMIT SWITCH SIGNAL TO MICROPROCESSOR
- (9) DOWN LIMIT SWITCH SIGNAL TO MICROPROCESSOR
- (10) WEST LIMIT SWITCH SIGNAL TO MICROPROCESSOR
- (11) NOT USED
- (12) +15 VOLTS TO LM 747 CHIP
- (13) -15 VOLTS TO LM 747 CHIP
- (14) GROUND

JACK J2

- (1) UP SENSOR INPUT FROM TRACKER
- (2) EAST SENSOR INPUT FROM TRACKER
- (3) VERTICAL DRIVE STEP ELEMENT #1 FROM MICROPROCESSOR
- (4) VERTICAL DRIVE STEP ELEMENT #2 FROM MICROPROCESSOR
- (5) VERTICAL DRIVE STEP ELEMENT #3 FROM MICROPROCESSOR
- (6) VERTICAL DRIVE STEP ELEMENT #4 FROM MICROPROCESSOR
- (7) GROUND
- (8) NOT USED
- (9) WEST SENSOR INPUT FROM TRACKER
- (10) DOWN SENSOR INPUT FROM TRACKER
- (11) HORIZONTAL DRIVE STEP ELEMENT #1 FROM MICROPROCESSOR
- (12) HORIZONTAL DRIVE STEP ELEMENT #2 FROM MICROPROCESSOR
- (13) HORIZONTAL DRIVE STEP ELEMENT #3 FROM MICROPROCESSOR
- (14) HORIZONTAL DRIVE STEP ELEMENT #4 FROM MICROPROCESSOR

JACK J3

- (1) +6.3 VOLTS TO VERTICAL STEPPING MOTOR
- (2) AMPLIFIED VERTICAL STEP ELEMENT #3 TO VERTICAL STEPPING MOTOR
- (3) +6.3 VOLTS TO VERTICAL STEPPING MOTOR
- (4) AMPLIFIED VERTICAL STEP ELEMENT #1 TO VERTICAL STEPPING MOTOR

- (5) AMPLIFIED VERTICAL STEP ELEMENT #2 TO VERTICAL STEPPING MOTOR
- (6) AMPLIFIED VERTICAL STEP ELEMENT #4 TO VERTICAL STEPPING MOTOR
- (7) AMPLIFIED HORIZONTAL STEP ELEMENT #4 TO HORIZONTAL STEPPING MOTOR
- (8) +6.3 VOLTS TO HORIZONTAL STEPPING MOTOR
- (9) +6.3 VOLTS TO HORIZONTAL STEPPING MOTOR
- (10) AMPLIFIED HORIZONTAL STEP ELEMENT #3 TO HORIZONTAL STEPPING MOTOR
- (11) AMPLIFIED HORIZONTAL STEP ELEMENT #2 TO HORIZONTAL STEPPING MOTOR
- (12) AMPLIFIED HORIZONTAL STEP ELEMENT #1 TO HORIZONTAL STEPPING MOTOR
- (13) NOT USED
- (14) NOT USED

SINGLE PIN JACKS

- (A) +5 VOLTS
- (B) +5 VOLTS TO LIMIT SWITCHES
- (C) -15 VOLT INPUT WHEN NOT USING INTEL 80/10A
- (D) +15 VOLT INPUT WHEN NOT USING INTEL 80/10A
- (E) EAST LIMIT SWITCH SIGNAL FROM TRACKER
- (F) NOT USED
- (G) WEST LIMIT SWITCH SIGNAL FROM TRACKER
- (H) NOT USED
- (I) NOT USED
- (J) +6.3 VOLTS FROM EXTERNAL POWER SUPPLY
- (K) +6.3 VOLTS FROM EXTERNAL POWER SUPPLY
- (L) GROUND
- (M) GROUND
- (N) NOT USED
- (O) NOT USED
- (P) UP LIMIT SWITCH SIGNAL FROM TRACKER
- (Q) NOT USED
- (R) DOWN LIMIT SWITCH SIGNAL FROM TRACKER
- (S) NOT USED
- (T) NOT USED
- (U) GROUND
- (V) GROUND

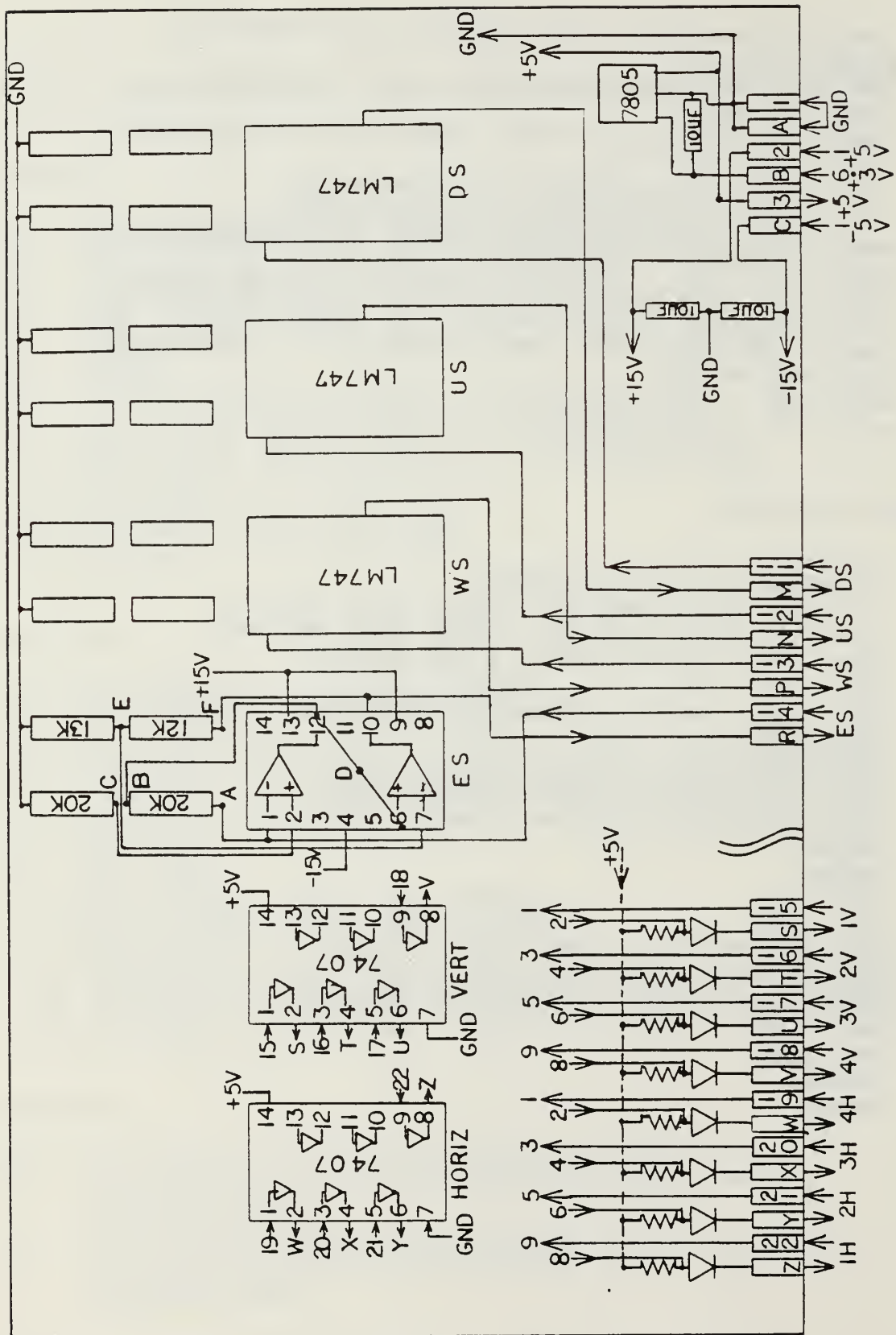


Figure 7. Schematic Diagram of Circuit Board Inside Amplification/Isolation Box

TABLE 2

AMPLIFICATION/ISOLATION BOX CIRCUIT BOARD SIGNALS
(LETTERED PINS ARE ON TOP OF CIRCUIT BOARD)

- (A) GROUND
- (1) GROUND
- (B) +6.3 VOLTS TO 7805 VOLTAGE REGULATOR FOR +5 VOLTS
REGULATED SUPPLY TO 7407 CHIPS
- (2) +15 VOLTS IN FOR LM 747 CHIPS
- (C) -15 VOLTS IN FOR LM 747 CHIPS
- (3) +5 VOLTS OUT USED BY LIMIT SWITCHES
- (M) DOWN SENSOR AMPLIFIED SIGNAL OUTPUT
- (11) DOWN SENSOR UNAMPLIFIED SIGNAL INPUT TO LM 747 CHIP
- (N) UP SENSOR AMPLIFIED SIGNAL OUTPUT
- (12) UP SENSOR UNAMPLIFIED SIGNAL INPUT TO LM 747 CHIP
- (P) WEST SENSOR AMPLIFIED SIGNAL OUTPUT
- (13) WEST SENSOR UNAMPLIFIED SIGNAL INPUT TO LM 747 CHIP
- (R) EAST SENSOR AMPLIFIED SIGNAL OUTPUT
- (14) EAST SENSOR UNAMPLIFIED SIGNAL INPUT TO LM 747 CHIP
- (S) VERTICAL STEP ELEMENT #1 OUTPUT TO POWER TRANSISTOR
BASE
- (15) VERTICAL STEP ELEMENT #1 INPUT FROM MICROPROCESSOR
- (T) VERTICAL STEP ELEMENT #2 OUTPUT TO POWER TRANSISTOR
BASE
- (16) VERTICAL STEP ELEMENT #2 INPUT FROM MICROPROCESSOR
- (U) VERTICAL STEP ELEMENT #3 OUTPUT TO POWER TRANSISTOR
BASE
- (17) VERTICAL STEP ELEMENT #3 INPUT FROM MICROPROCESSOR
- (V) VERTICAL STEP ELEMENT #4 OUTPUT TO POWER TRANSISTOR
BASE
- (18) VERTICAL STEP ELEMENT #4 INPUT FROM MICROPROCESSOR
- (W) HORIZONTAL STEP ELEMENT #4 OUTPUT TO POWER TRANSISTOR
BASE
- (19) HORIZONTAL STEP ELEMENT #4 INPUT FROM MICROPROCESSOR
- (X) HORIZONTAL STEP ELEMENT #3 OUTPUT TO POWER TRANSISTOR
BASE
- (20) HORIZONTAL STEP ELEMENT #3 INPUT FROM MICROPROCESSOR
- (Y) HORIZONTAL STEP ELEMENT #2 OUTPUT TO POWER TRANSISTOR
BASE
- (21) HORIZONTAL STEP ELEMENT #2 INPUT FROM MICROPROCESSOR
- (Z) HORIZONTAL STEP ELEMENT #1 OUTPUT TO POWER TRANSISTOR
BASE
- (22) HORIZONTAL STEP ELEMENT #1 INPUT FROM MICROPROCESSOR

figure. Additionally all signals entering and leaving the circuit board apply to either the Intel or Apple computers. Figure 8 shows the remainder of the electronics inside the A/I box, namely the eight power transistors used to amplify the step signals sent to the stepping motors. The step elements V1 through V4 and H1 through H4 will be explained in detail later. Figure 9 and Table 3 are included only for completeness of system documentation, and to avoid future duplication of effort. They represent the Intel 80/10A wiring and signal description.

C. SYSTEM OPERATION

Once the signal flow had been determined, the actual system operation was the next focus of investigation. This information was necessary because the characteristics of the signals coming into, and the type of the signals going out of the new computer had to be known.

There were two areas of investigation during the system operation phase. The first was the two driver stepping motors, and their control circuitry. A thorough understanding of their operation was essential to the later development of the software drive routines. As no technical documentation was readily available for the stepping motors that were used, the following data was obtained from [Ref. 2], experimentation, name plate data, and general stepping motor theory [Ref. 3]. The motors used were Superior Electric synchronous

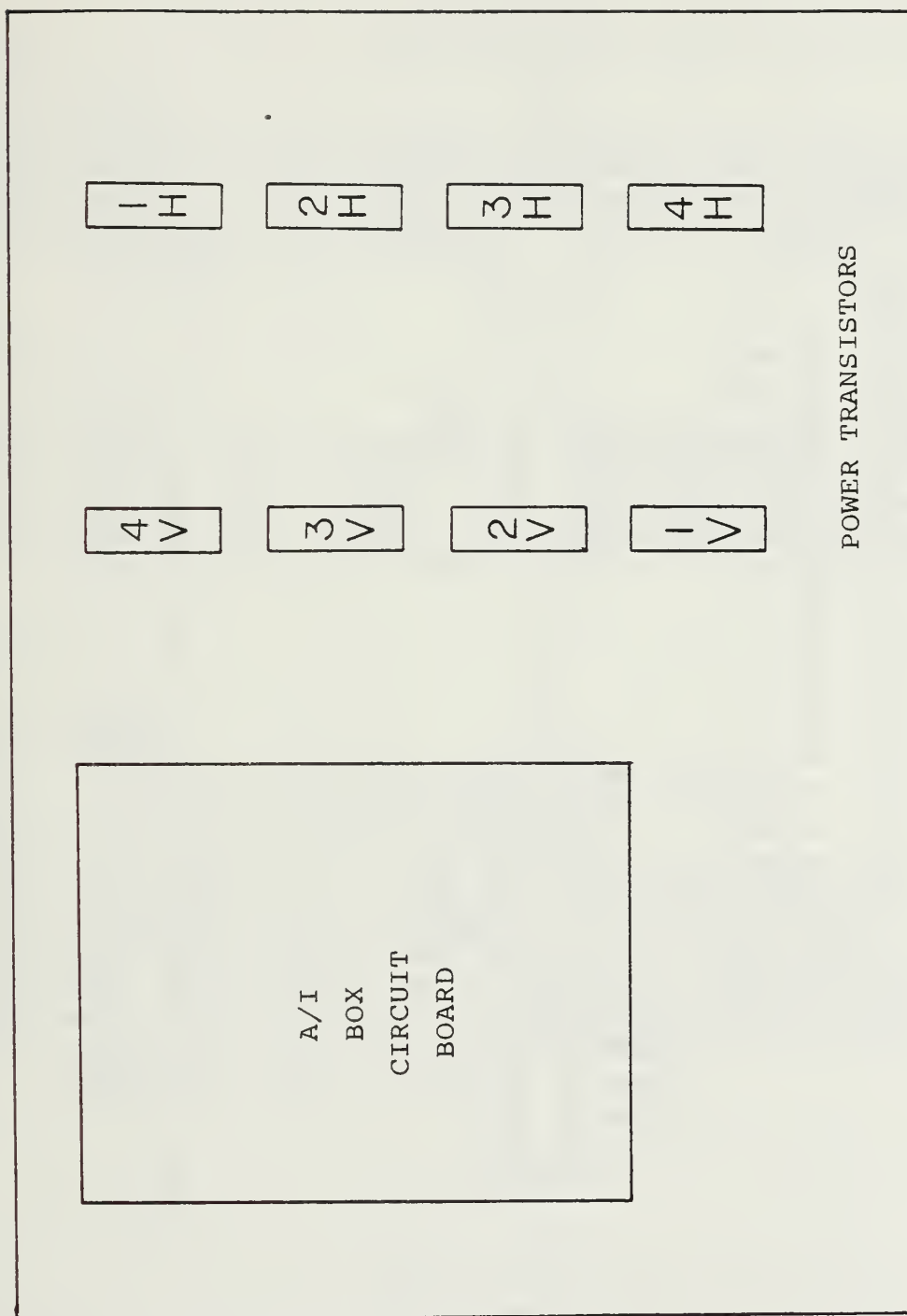


Figure 8. Power Transistor Layout Inside Amplification/Isolation Box

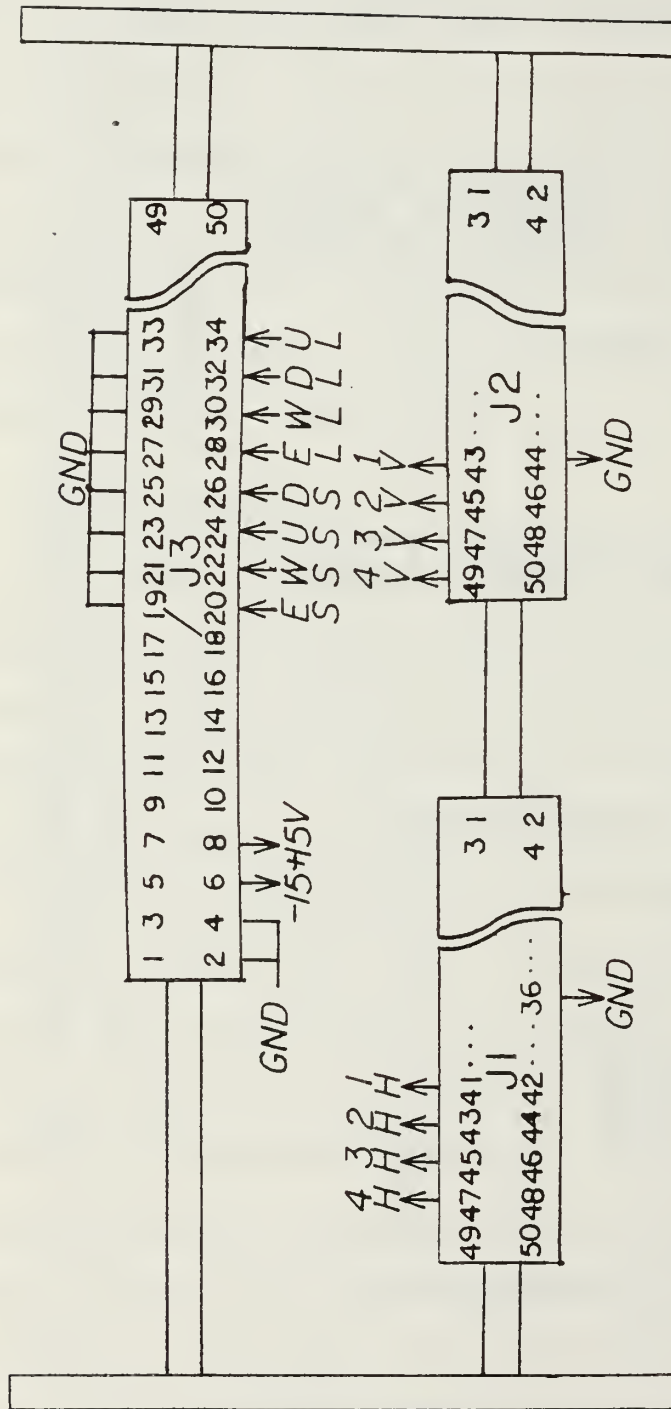


Figure 9. INTEL 80/10A Single Board Computer Connections

TABLE 3

INTEL 80/10A SINGLE BOARD COMPUTER CONNECTIONS

JACK J1

- (36) GROUND
- (41) HORIZONTAL STEP ELEMENT #1 TO AMPLIFIER/ISOLATION BOX
- (43) HORIZONTAL STEP ELEMENT #2 TO AMPLIFIER/ISOLATION BOX
- (45) HORIZONTAL STEP ELEMENT #3 TO AMPLIFIER/ISOLATION BOX
- (47) HORIZONTAL STEP ELEMENT #4 TO AMPLIFIER/ISOLATION BOX
- (REMAINING PINS NOT USED)

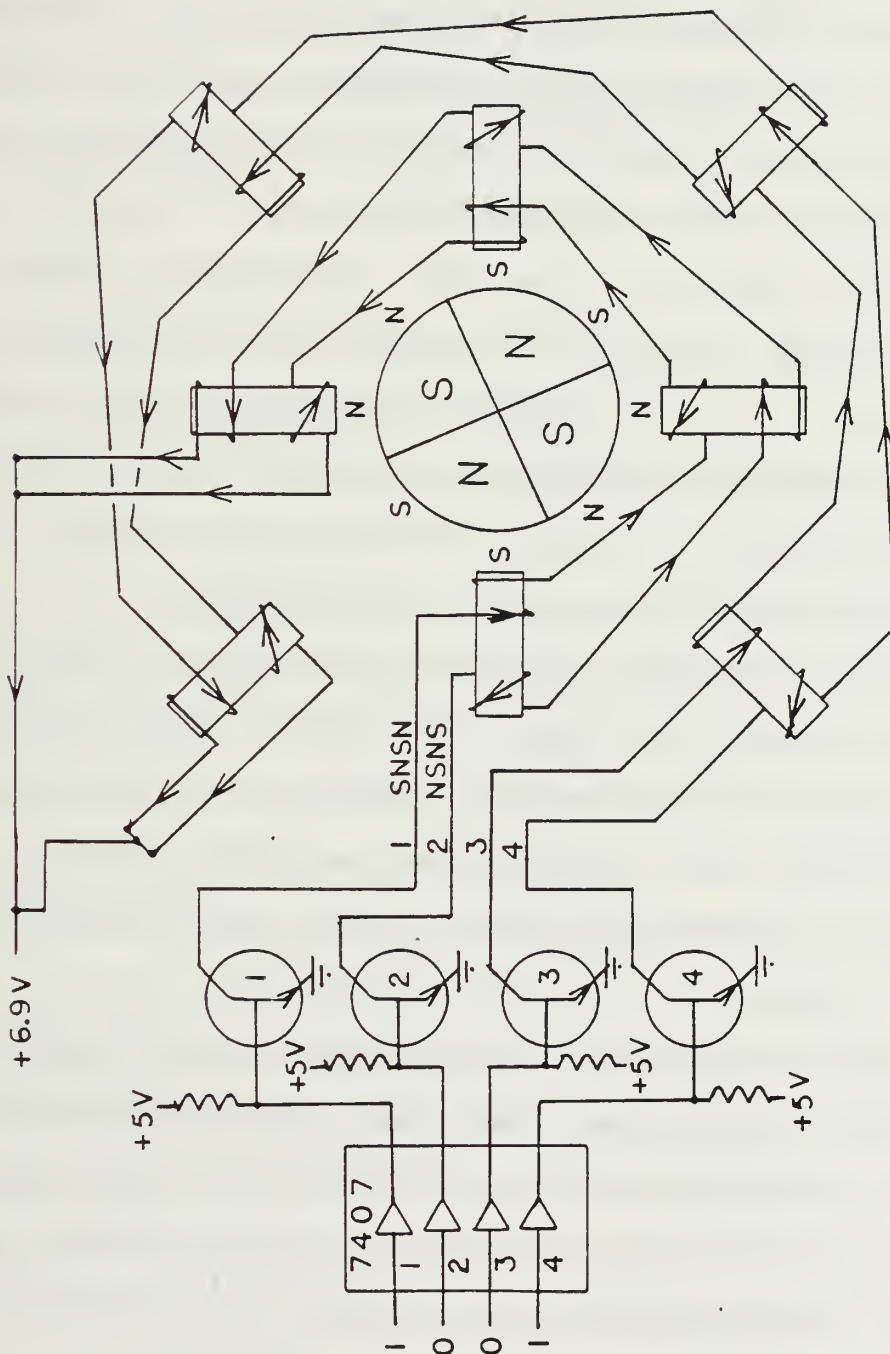
JACK J2

- (44) GROUND
- (43) VERTICAL STEP ELEMENT #1 TO AMPLIFIER/ISOLATION BOX
- (45) VERTICAL STEP ELEMENT #2 TO AMPLIFIER/ISOLATION BOX
- (47) VERTICAL STEP ELEMENT #3 TO AMPLIFIER/ISOLATION BOX
- (49) VERTICAL STEP ELEMENT #4 TO AMPLIFIER/ISOLATION BOX
- (REMAINING PINS NOT USED)

JACK J3

- (2) GROUND
- (4) GROUND
- (6) -15 VOLTS TO AMPLIFIER/ISOLATION BOX
- (8) +15 VOLTS TO AMPLIFIER/ISOLATION BOX
- (18, 19, 21, 23, 25, 27, 29, 31, 33) GROUND
- (20) EAST SENSOR INPUT FROM AMPLIFIER/ISOLATION BOX
- (22) WEST SENSOR INPUT FROM AMPLIFIER/ISOLATION BOX
- (24) UP SENSOR INPUT FROM AMPLIFIER/ISOLATION BOX
- (26) DOWN SENSOR INPUT FROM AMPLIFIER/ISOLATION BOX
- (28) EAST LIMIT SWITCH INPUT FROM AMPLIFIER/ISOLATION BOX
- (30) WEST LIMIT SWITCH INPUT FROM AMPLIFIER/ISOLATION BOX
- (32) DOWN LIMIT SWITCH INPUT FROM AMPLIFIER/ISOLATION BOX
- (REMAINING PINS NOT USED)

stepping motors that have a nominal DC voltage rating of 6.9 Volts and a rated amperes per winding value of 1.6 Amps. Figure 10 is a schematic representation of a typical stepping motor, and its interface to the system. In this schematic the rotor is a four pole permanent magnet (large circle). There are eight stators, each of which is an electromagnet. These stators are divided into two groups of four. One group is controlled by power transistors 1 and 2, the other by 3 and 4. Each stator is wound by wire in two directions. For the set controlled by power transistors 1 and 2 the polarities of each stator depend on whether transistor 1 or 2 is turned on. The left hand rule for electron current flow can be used to determine the polarity of each sensor. By wrapping the fingers of the left hand around the stator in the direction of current flow, the north pole is determined to be the end of the stator that the thumb points to. For example if transistor 1 is turned on, electron current will flow from the grounded terminal of the transistor to the +6.9 Volts at the top of the diagram. By applying the left hand rule, a S-N-S-N configuration is observed. If transistor 2 were conducting a N-S-N-S configuration would result. Transistors 1 and 2 conducting at the same time is not a valid state. Transistors 3 and 4 control the other set of stators in a similar manner. It is by this ability to alter the polarity of these stator sets that the stepping motion of the motors is realized. An example step command is illustrated in the



(arrows indicate the direction of electron current flow)

Figure 10. Schematic Diagram of Typical Stepping Motor

figure. In this instance a 1001 step is input to a 7407 noninverting hex driver chip. A "1" represents a +5 Volt high signal, and a "0" represents a zero Volt low signal. When a low signal is applied, the driver can sink up to 30mA., and the +5 volts at the base of the transistor is dropped across the base resistor thus keeping the transistor turned off. When a high signal is sensed the driver becomes an open circuit, and the +5 Volts is no longer dropped across the base resistor. The +5 Volts is sensed at the base of the NPN transistor, and it is turned on. Current can now be conducted through the stator windings so that the stators are polarized. With the above description in mind, the 1001 step example can easily be shown to produce the indicated stator polarizations. Finally remembering that unlike poles attract, the rotor orientation is set as shown.

In summary each step command to the stepping motor has four step elements, as in the 1001 example. The first two elements control one set of four stators, and the last two steps the second set. The polarity configuration of a stator set is determined by which of the two controlling step elements is high.

The above description is for a four pole rotor with eight stators. The actual stepping motor used has four 50 pole rotors stacked one above the other, each offset slightly from the one below it. This offset in essence gives a 100 pole rotor. There are still only eight electromagnetic

stators, but each has five high points on its surface. The magnetic field is concentrated at the high points giving the appearance of five stators of the same polarity. This large number of poles allows the step size to be either 0.9 or 1.8 degrees, depending upon the drive sequence used to step the motor. These step sizes correspond to either 400 or 200 steps per 360 degree revolution. The example that was discussed in Figure 10 would have produced 22.5 or 45 degree steps for 16 or 8 steps per revolution, again depending upon the drive sequence used. The smaller steps of the real motor are clearly more useful in real world applications than the large steps of the example.

As mentioned the step size depends on the drive sequence used. Figure 11 shows a four step loop. Notice that each change in stator polarization produces a 45 degree movement of the rotor. The loop is repeated over and over to drive the stepping motor the desired number of steps. In this case the loop must be repeated twice per 360 degree revolution. It should be noted that step one produced a 45 degree movement of the rotor to place it in the first position, so that four steps produce a 180 degree transition. The small numbers 1 and 2 on the first step, mark the starting positions of the first and second stator sets. The stators are laid out in the same manner as those in Figure 10. Again the first two elements of each step control stator set one, and the last two stator set two. Figure 12 is a typical eight step loop.

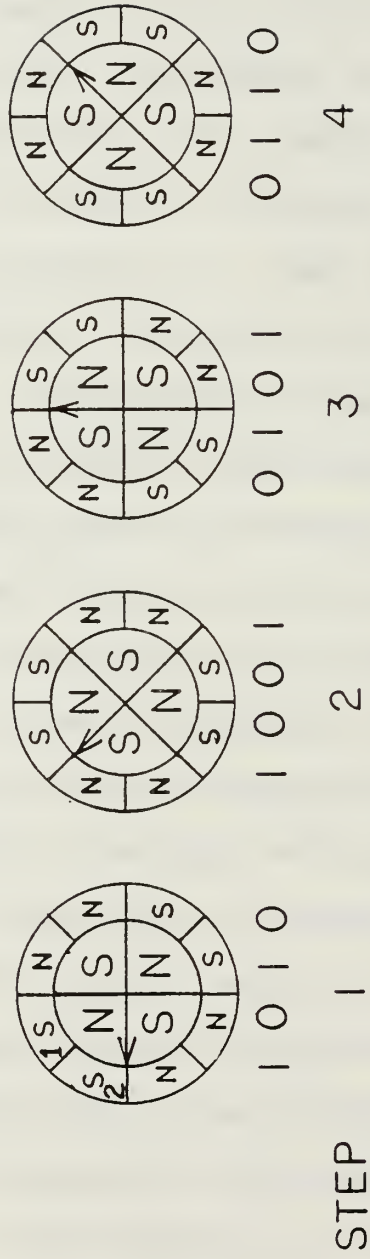


Figure 11. Four Step Drive Routine

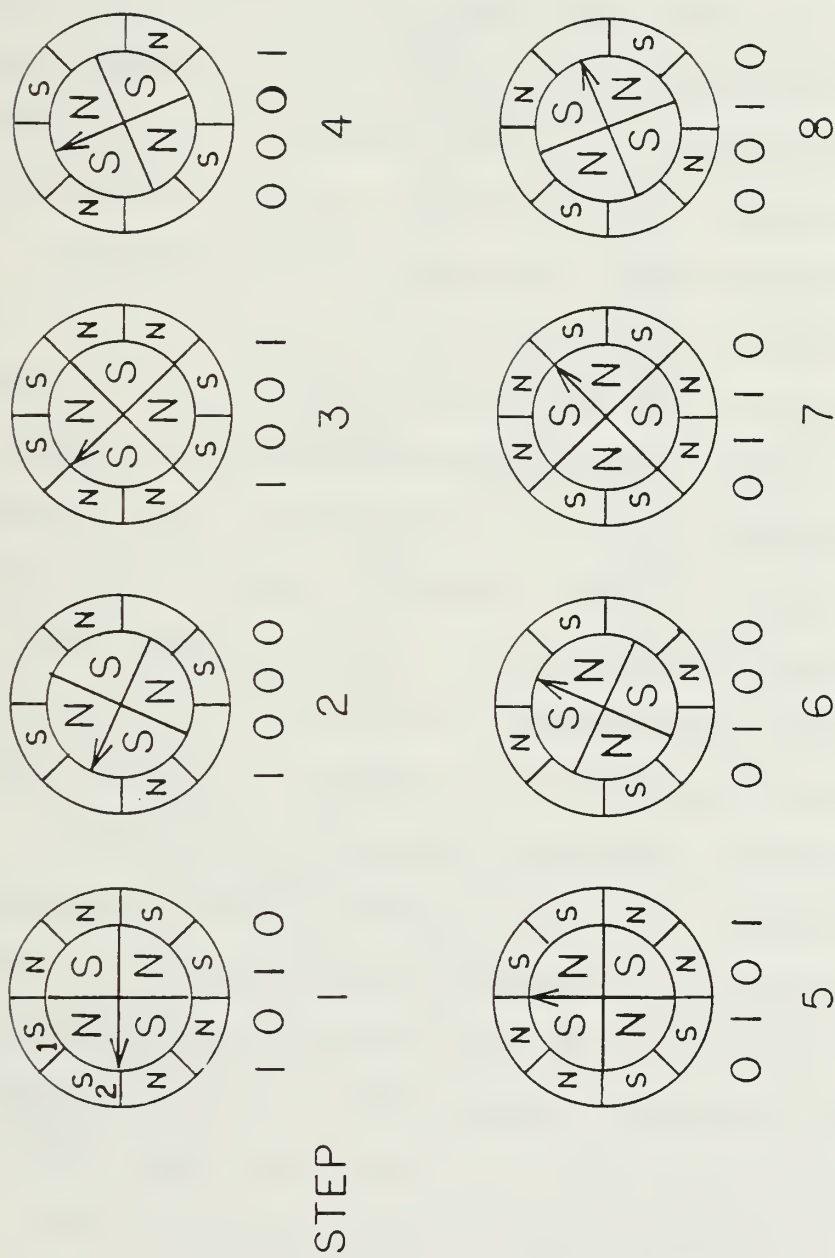


Figure 12. Eight Step Drive Sequence

In this case each step moves the rotor 22.5 degrees. The smaller steps are realized by inserting four extra steps between the original four. In each of the extra steps one of the stator sets is left off by not turning on either of its associated transistors. The rotor aligns itself as illustrated, on the energized stator set. It is these same 4 and 8 step loops that produce the 0.9 and 1.8 degree steps of the real drive motors. The only difference is that the loop must be repeated many more times per motor revolution. With the above information, drive routines for the new computer could now be developed.

The second of the two main areas of investigation was the sensor circuitry. As mentioned the sensors are photodiodes that generate a current proportional to the intensity of the light falling on them. Figure 13 shows sensor output signal amplification circuitry for the east sensor. The letters A through F correspond to the letters at the top of the A/I box circuit board shown in Figure 7, so that this figure can be tied to the actual hardware. The circuit operates as follows. The first op amp provides a voltage output proportional to the current input from the energized photodiode. This voltage is then further amplified by the second op amp, to provide a signal that is usable by the computer. This output varies from near zero volts when the sensor is in a shadow, to near 2.5 volts in the bright sunlight.

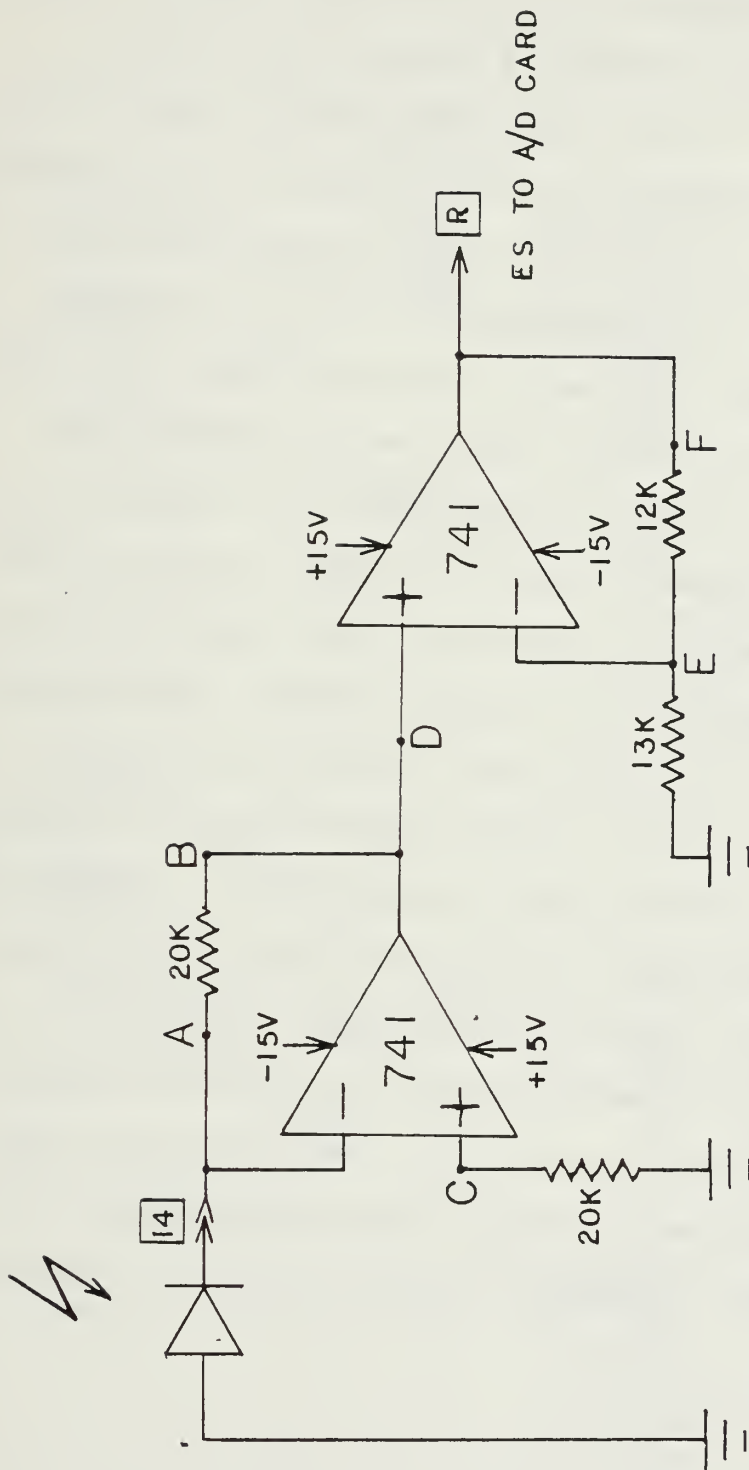


Figure 13. Sensor Signal Amplification Circuit

D. COMPUTER INTEGRATION

With the preliminary research complete, the problem of integrating the new computer into the system could be taken up. The Apple II Plus computer is equipped with several internal card slots, so that integrated circuit cards which perform many different functions can be added to the computer. After some difficulty in choosing a card to use for interfacing the computer to the system, an analog to digital/digital to analog card (A/D card) was chosen [Ref. 4]. This card has 16 channels of analog input and 16 channels of analog output. Each of the 32 channels has an address and can be accessed by software commands. Eight of the digital to analog output channels (4 per motor) were used to deliver the drive step commands to the stepper motors. Additionally eight of the analog to digital input channels were used to input the 4 sensor and the 4 limit switch signals. Figure 14 shows the A/D card and the jacks coming from it. These jacks are connected to two similar jacks coming from the A/I box. Table 4 describes the signals on the jacks, lists the A/D card channel associated with each pin (whether used or not), and gives the color of the wire associated with each signal for ease of identification. With the hardware interfacing completed the development of the controlling software could begin.

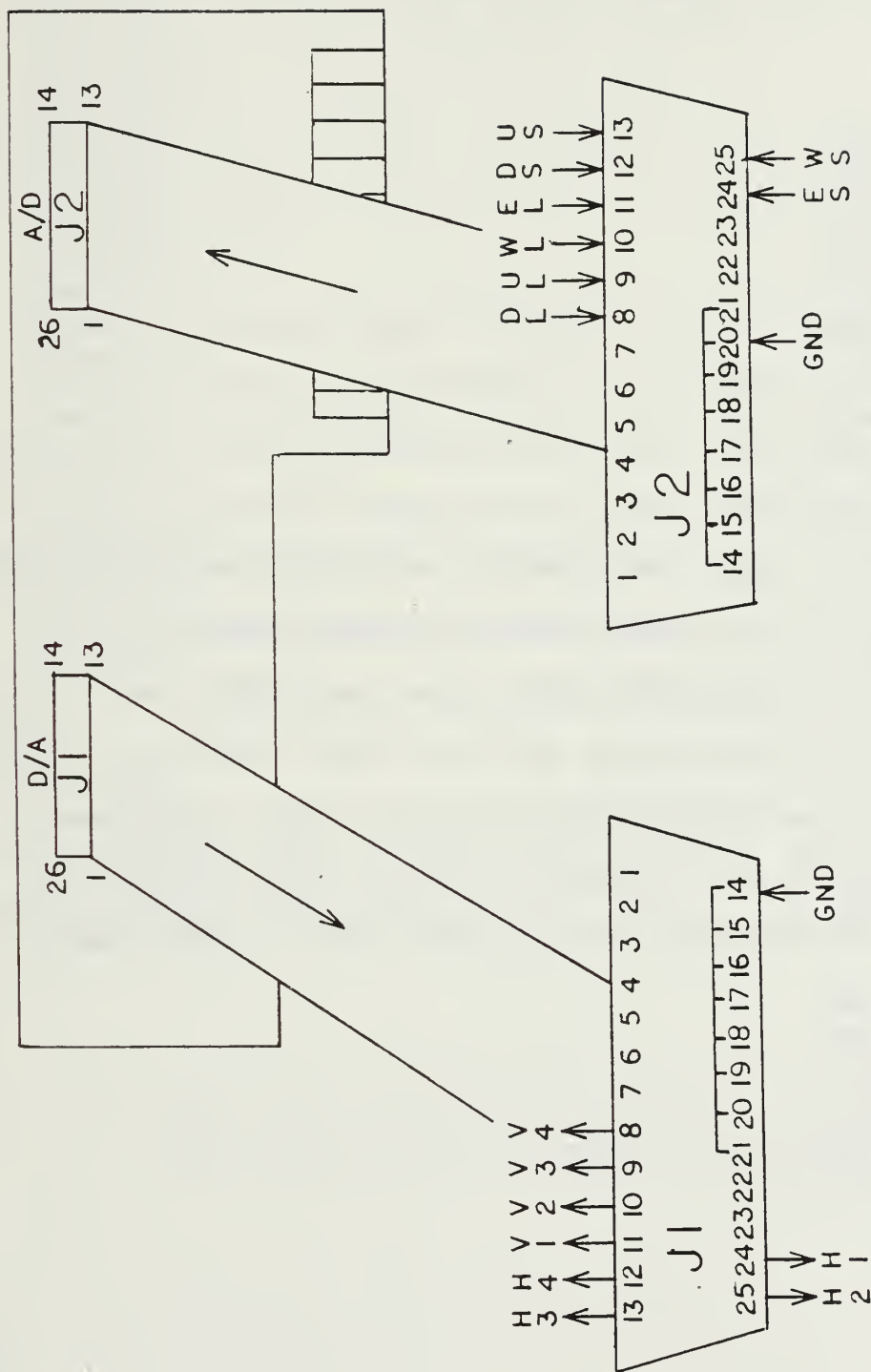


Figure 14. Analog to Digital Card with Connecting Jacks

TABLE 4

LISTING OF THE A/D CARD JACK CONNECTIONS
(WIRE COLOR GIVEN AT END OF SIGNAL DESCRIPTION)
(NOTE CARD CHANNEL GIVEN BELOW SIGNAL DESCRIPTION)

JACK #1

- (1) NOT USED
CHANNEL 15
- (2) NOT USED
CHANNEL 14
- (3) NOT USED
CHANNEL 13
- (4) NOT USED
CHANNEL 12
- (5) NOT USED
CHANNEL 11
- (6) NOT USED
CHANNEL 10
- (7) NOT USED
CHANNEL 9
- (8) VERTICAL STEP ELEMENT #4 TO AMP/ISOLATION BOX (YELLOW)
CHANNEL 8
- (9) VERTICAL STEP ELEMENT #3 TO A/I BOX (GREEN)
CHANNEL 7
- (10) VERTICAL STEP ELEMENT #2 TO A/I BOX (RED)
CHANNEL 6
- (11) VERTICAL STEP ELEMENT #1 TO A/I BOX (BLUE)
CHANNEL 5
- (12) HORIZONTAL STEP ELEMENT #4 TO A/I BOX (YELLOW)
CHANNEL 4
- (13) HORIZONTAL STEP ELEMENT #3 TO A/I BOX (GREEN)
CHANNEL 3
- (14-21) COMMON GROUNDS, GROUND WIRE FROM A/I BOX IN 14
(BLACK)
- (22) NOT USED
- (23) NOT USED
CHANNEL 0
- (24) HORIZONTAL STEP ELEMENT #1 TO A/I BOX (BLUE)
CHANNEL 1
- (25) HORIZONTAL STEP ELEMENT #2 TO A/I BOX (RED)
CHANNEL 2

JACK #2

- (1) NOT USED
CHANNEL 15
- (2) NOT USED
CHANNEL 14
- (3) NOT USED
CHANNEL 13
- (4) NOT USED
CHANNEL 12
- (5) NOT USED
CHANNEL 11
- (6) NOT USED
CHANNEL 10
- (7) NOT USED
CHANNEL 9
- (8) DOWN LIMIT SWITCH INPUT FROM A/I BOX (YELLOW)
CHANNEL 8
- (9) UP LIMIT SWITCH INPUT FROM A/I BOX (GREEN)
CHANNEL 7
- (10) WEST LIMIT SWITCH INPUT FROM A/I BOX (RED)
CHANNEL 6
- (11) EAST LIMIT SWITCH INPUT FROM A/I BOX (BLUE)
CHANNEL 5
- (12) DOWN SENSOR INPUT FROM A/I BOX (YELLOW)
CHANNEL 4
- (13) UP SENSOR INPUT FROM A/I BOX (GREEN)
CHANNEL 3
- (14-21) COMMON GROUNDS, GROUND WIRE FROM A/I BOX IN 20
(BLACK)
- (22) NOT USED
- (23) NOT USED
CHANNEL 0
- (24) EAST SENSOR INPUT FROM A/I BOX (BLUE)
CHANNEL 1
- (25) WEST SENSOR INPUT FROM A/I BOX (RED)
CHANNEL 2

III. SOFTWARE DESIGN

A. INITIAL PLANNING

The controlling software for the system was developed in three distinct phases. The first phase was the design of the driver routines for the system stepping motors. The second phase was the development of the sensor and limit switch inputs. The third phase was to combine the first two phases into a program that would control the system in the desired manner. The program developed in the last phase had to be capable of obtaining the input data from the 4 photo-diode sensors, analyzing it, and causing the appropriate drive motor to turn in the proper direction. The program must also ensure that none of the limit switches had been depressed, and react accordingly if one had been.

B. DRIVE ROUTINE DESIGN

As was mentioned previously, the Apple computer has several internal slots available for the installation of various cards. Since each drive step consists of 4 bits of information, a parallel I/O card was initially chosen to output the step information to the drive motors. Specifically an A488 Communication Interface card was chosen [Ref. 5]. This card is designed for communication between the Apple and digitally controlled peripherals. The card has 8 bi-directional data lines, and 8 control lines. The data sent

and received by the card is by means of the ASCII character set. The control lines of the card are used for interdevice synchronization and data transmission setup commands.

Attempts were made to utilize this card by having it output the ASCII character whose last four bits corresponded to the desired drive step. Figure 15 is an example of a program developed for an 8 step drive routine. Lines 70 through 170 are setup commands required by the card, and 180 through 370 are the drive loop. All attempts to make this card work failed due to the lack of handshaking and control signals coming from the seeker system.

The A/D card mentioned in the last section was also in the computer, and was to be used for the conversion of the analog input signals into a digital form usable by the computer. The fact that the A/D card also had 16 output channels lead to the idea that these channels could be used to produce the necessary drive steps. It was concluded that if the 4 elements comprising the step arrived at the stepping motor in rapid serial succession rather than at the same instant in parallel form, the stepping motor would still perform satisfactorily. This would allow for the use of only one card for the complete system interface, rather than the two required if the A488 were used.

The value of a particular A/D card output channel is set by sending a number from 0 to 255 to the channel address. The following are some of the output voltage values corresponding to various numbers sent to the channel.

```

*** PROGRAM TO DRIVE SEEKER WITH ***
*** A488 PARALLEL I/O CARD ***

```

```

10  SL% = 5
20  DT$ = "SUN:01"
30  RS$ = ""
40  FS$ = ""
50  CI$ = ""
60  PA% = 1
70  X = 256 * SL%
80  IC = 49152 + X
90  SN = IC + 16
100 FI = IC + 32
110 PA = IC + 96
120 SO = IC + 192
130 SR = 49295 + SL% * 16
140 PE = 1144 + SL%:DU = 1272 + SL%
150 EOS = 1912 + SL%
160 CALL FI:
170 CALL IC:
180 FOR T = 1 TO 16:RS$ = RS$ + "": NEXT
190 FOR A = 1 TO 100
200 CI$ = "S,SUN":FS$ = "J": CALL SN:
210 PA% = 100: CALL PA:
220 CI$ = "S,SUN":FS$ = "H": CALL SN:
230 PA% = 100: CALL PA:
240 CI$ = "S,SUN":FS$ = "I": CALL SN:
250 PA% = 100: CALL PA:
260 CI$ = "S,SUN":FS$ = "A": CALL SN:
270 PA% = 100: CALL PA:
280 CI$ = "S,SUN":FS$ = "E": CALL SN:
290 PA% = 100: CALL PA:
300 CI$ = "S,SUN":FS$ = "D": CALL SN:
310 PA% = 100: CALL PA:
320 CI$ = "S,SUN":FS$ = "L": CALL SN:
330 PA% = 100: CALL PA *
340 CI$ = "S,SUN":FS$ = "B": CALL SN:
360 PA% = 100: CALL PA:
370 NEXT A
380 END

```

Figure 15. A488 Drive Routine

| NUMBER | OUTPUT |
|--------|------------|
| 0 | -5.0 Volts |
| 64 | -2.5 |
| 128 | 0.0 |
| 192 | +2.5 |
| 255 | +5.0 |

From the table it can be seen that the card can output from -5 to +5 volts in .039 Volt increments. The values 128 and 255 were chosen to represent the 0 and the 1 values for each step element. Analog output channels 1-4 were used for the 4 horizontal drive step elements, and 5-8 for the vertical step elements (see Table 4). To define a particular output from a channel, the base channel address must first be computed. To do this the equation $M = 49,280 + (\text{slot number} * 16)$ is used. Here the 49,280 is the starting address of all slot dependent locations in the computer, and the slot number is an integer from 1 to 7, depending upon the slot that the card was installed in. Next a channel number from 0 to 15 must be added to the base address M in order to address a particular channel on the card. Note that the A/D card must be installed in slot number four if the sun seeker program is to work properly. Once the A/D card channel address ($M + \text{CHANNEL ADDRESS}$) has been calculated, the POKE command can be used to input any value to the channel within the allowable range. POKE is a BASIC command which has the form `POKE(memory address), (desired value)`. Figure

16 is an early drive routine using the POKE command. The slot number that the card is installed in is given in line 10, lines 20-45 assign step element values to an array, and lines 50-120 are three nested Do Loops. The inner most Do Loop I, calculates the memory base value M, and POKES the values of the indicated array element into the proper card channel (M+I). So the basic memory value M is incremented by adding the Do Loop values I (1-4) to it, thus four channel addresses are calculated. These addresses correspond to the A/D card channels used to output the 4 element drive step. These addresses together with the desired output voltage values are placed in POKE commands to form the step. The next Do Loop in Figure 16 (the H loop), generates a four step loop like that of Figure 11. Finally the G loop causes the 4 step drive loop to be repeated 60 times.

When this routine was run it suddenly became clear that great programming efficiency would be necessary if any kind of acceptable system response times were to be achieved. To make a transition of only 90 degrees with this drive routine, 25 seconds were required. In other words to move the tracker from an east to a west facing position, it would take close to one minute. Figure 17 is the same drive routine as that of Figure 16, except that the M calculation was taken out of the Do Loop, and calculated only once at the beginning of the program. This simple change reduced the 90 degree transition time to 12 seconds, for a savings of 13 seconds. While this


```

*** FIRST DRIVE ROUTINE USING A/D ***
*** CARD. MEMORY CALCULATION IN LOOP***
***   REQUIRED 25 SEC. FOR 90 DEG   ***
***           TRANSITION.           ***

```

```

10  SL = 4
20  DIM D(16)
30  D(1) = 255:
    D(2) = 128:
    D(3) = 255:
    D(4) = 128
35  D(5) = 255:
    D(6) = 128:
    D(7) = 128:
    D(8) = 255
40  D(9) = 128:
    D(10) = 255:
    D(11) = 128:
    D(12) = 255
45  D(13) = 128:
    D(14) = 255:
    D(15) = 255:
    D(16) = 128
50  FOR G = 1 TO 60
60  FOR H = 1 TO 4
70  FOR I = 1 TO 4
75  M = 49280 + (SL * 16)
80  POKE M + I, D(I + (4 * (H - 1)))
90  NEXT
110 NEXT
120 NEXT

```

Figure 16. 25 Second Drive Routine

```

***      SECOND DRIVE ROUTINE.      ***
***      BY MOVING MEMORY CALC.     ***
***      OUTSIDE LOOP, REDUCED TRANSIT ***
***      TIME TO 12.4 SEC FOR A 90 DEG ***
***      TRANSITION.                ***

```

```

10  SL = 4
15  M = 49280 + (SL * 16)
20  DIM D(16)
30  D(1) = 255:
    D(2) = 128:
    D(3) = 255:
    D(4) = 128
35  D(5) = 255:
    D(6) = 128:
    D(7) = 128:
    D(8) = 255
40  D(9) = 128:
    D(10) = 255:
    D(11) = 128:
    D(12) = 255
45  D(13) = 128:
    D(14) = 255:
    D(15) = 255:
    D(16) = 128
50  FOR G = 1 TO 60
60  FOR H = 1 TO 4
70  FOR I = 1 TO 4
80  POKE M + I, D(I + (4 * (H - 1)))
90  NEXT
110 NEXT
120 NEXT

```

Figure 17. 12 Second Drive Routine

change cut the time in half, still more time would have to be cut because in actual use the drive routine would have logic placed between the steps, further slowing it down.

Figure 18 shows the third drive routine developed. To increase execution speed the step elements were taken out of an array, and placed directly into the poke commands. The A/D card channel addresses for the 4 channels used per step are calculated at the beginning of the program before entering the drive loop, and are also placed directly into the poke commands. Variable names vice constant values were used in the poke commands to further increase speed. This is because it takes much more time to convert a constant to its floating point representation than it does to fetch the value of a variable [Ref. 6]. Due to the above modifications the 90 degree transition time was reduced to 4.8 seconds. Figure 19 is the same type drive routine as that of Figure 18 except that a four step drive loop is used. As would be expected the transit time was cut in half to 2.6 seconds due to the larger step size, however the operation of the drive motor became unstable. At this speed the time to serially deliver the four elements comprising a step approaches the time between steps of 5 msec. It is therefore possible for the inertia of the larger steps (1.9 vs. 0.8 degrees), to carry the rotor past the next intended position before the new step is set up. The drive motor would operate normally for a time and then suddenly begin to oscillate about some position.

```

***  THIRD DRIVE DESIGN USING A/D  ***
***  CARD. HERE EACH DRIVE STEP HAS ***
***  MEMORY ADDRESS ASSIGNED VICE ***
***  ACCESSING AN ARRAY AS IN FIRST ***
***  TWO DESIGNS. TRANSIT TIME FOR ***
***  90 DEG TRANSITION WAS REDUCED ***
***              TO 4.8 SEC.        ***

```

```

80      EN = 60
90      DE = 1
100     SL = 4
120     ML = 49280 + (SL * 16)
140     DIM M(4)
160     M1 = ML + 1
180     M2 = ML + 2
200     M3 = ML + 3
220     M4 = ML + 4
240     HI = 255
260     LO = 128
2000    FOR H1 = 1 TO EN
2100    POKE M1,HI:
          POKE M2,LO:
          POKE M3,HI:
          POKE M4,LO
2200    POKE M1,HI:
          POKE M2,LO:
          POKE M3,LO:
          POKE M4,LO
2300    POKE M1,HI:
          POKE M2,LO:
          POKE M3,LO:
          POKE M4,HI
2400    POKE M1,LO:
          POKE M2,LO:
          POKE M3,LO:
          POKE M4,HI
2500    POKE M1,LO:      2700 POKE M1,LO:
          POKE M2,HI:      POKE M2,HI:
          POKE M3,LO:      POKE M3,HI:
          POKE M4,HI      POKE M4,LO
2600    POKE M1,LO:      2800 POKE M1,LO:
          POKE M2,HI:      POKE M2,LO:
          POKE M3,LO:      POKE M3,LO:
          POKE M4,LO      POKE M4,HI

```

Figure 18. 4.8 Second Eight Step Drive Loop

```

***   THIS DRIVE DESIGN USES ONLY   ***
*** FOUR DRIVE STEPS PER LOOP. THE ***
*** TRANSITION TIME FOR THIS DESIGN ***
*** WAS ONLY 2.6 SEC, HOWEVER THE ***
*** DRIVE MOTOR OPERATION BECAME ***
***           UNRELIABLE.           ***

```

```

80     EN = 60
90     DE = 1
100    SL = 4
120    ML = 49280 + (SL * 16)
140    DIM M(4)
160    M1 = ML + 1
180    M2 = ML + 2
200    M3 = ML + 3
220    M4 = ML + 4
240    HI = 255
260    LO = 128
2000   FOR H1 = 1 TO EN
2100   POKE M1,LO:
        POKE M2,HI:
        POKE M3,HI:
        POKE M4,LO
2200   POKE M1,LO:
        POKE M2,HI:
        POKE M3,LO:
        POKE M4,HI
2300   POKE M1,HI:
        POKE M2,LO:
        POKE M3,LO:
        POKE M4,HI
2400   POKE M1,HI:
        POKE M2,LO:
        POKE M3,HI:
        POKE M4,LO
2900   NEXT

```

Figure 19. 2.6 Second Four Step Drive Loop

Figure 20 is a drive routine developed to take advantage of the fact that only one of the four step elements changes from one step to the next in an eight step loop. Only the step element that changed was sent to the drive motor. This routine also proved to be highly unstable and was discarded.

The eight step loop of Figure 18 was finally chosen for use in the main sun tracking program because of its speed, stability, and smoother operation. The four step loop even when stabilized by adding delays between the steps still produced a jerky drive motion, while the smaller 0.8 degree steps of the eight step loop caused the motion to smooth out.

C. DESIGN OF THE SIGNAL INPUT ROUTINES

To control the system the Apple computer must be capable of taking in the sensor and limit switch signals as well as driving the motors. To do this the PEEK command in conjunction with the A/D card is used. The analog voltage values of the signals are delivered directly to eight channels of the A/D card.

The card then changes the analog values into their digital equivalents for use by the computer. The range of digital voltage input values is the same as the output value range discussed earlier (0-255 digital for -5V to +5V analog). The PEEK command is of the form VARIABLE NAME = PEEK (A/D


```

***      IN THIS DRIVE DESIGN AN      ***
***      ATTEMPT WAS MADE TO TAKE     ***
***      ADVANTAGE OF THE FACT THAT   ***
***      ONLY ONE OF FOUR LEVELS CHANGE***
***      PER STEP. SO ONLY THE CHANGING***
***      ELEMENT IS POKED EACH STEP.  ***
***      THIS DESIGN PROVED TO BE     ***
***      COMPLETELY UNSTABLE AS A DRIVE***
***      ROUTINE.                      ***

```

```

390  FOR K = 1 TO 30
400  POKE 49344,255
405  POKE 49345,128
410  POKE 49346,255
415  POKE 49347,128
420  POKE 49346,128
425  POKE 49347,255
430  POKE 49344,128
435  POKE 49345,255
440  POKE 49347,128
445  POKE 49346,255
450  POKE 49345,128
460  NEXT K

```

Figure 20. Drive Routine Varying Single Step Element

CARD CHANNEL ADDRESS). The digital value found at the indicated address is assigned to the variable. With this particular A/D card the PEEK command must be given twice in order to get the correct value. The first time the command is issued the old value stored in the channel is returned, the second time the new sampled value is delivered. Figure 21 is a routine designed to input sensor and limit switch values, and display them on the system monitor.

In this routine the base address M is calculated in line 110 and incremented by the Do Loop counter I to get the eight input channel addresses of the A/D card. These are the same addresses as the eight used for outputting the drive steps. The command used (PEEK or POKE) determines whether the particular address is used for input or output. The input output wiring to and from the A/D card is physically separate (see Figure 14), it is only on the internal computer bus that the the same lines are used for input and output. The table below is included for a quick reference to the A/D card channel use.

| CARD ADDRESS | CHANNEL | INPUT | DRIVE STEP ELEMENT |
|--------------|---------|-------|-----------------------|
| USING SLOT 4 | NUMBER | USE | |
| 49345 | 01 | ES | H1 |
| 49346 | 02 | WS | H2 |
| 49347 | 03 | US | H3 |

```

***   THIS ROUTINE WAS THE FIRST   ***
***   ATTEMPT TO UTILIZE THE INPUT ***
***   SIDE OF THE A/D CARD. ALL    ***
***   SENSOR AND LIMIT SWITCH VALUES ***
***   ARE OBTAINED BY USING THE DO ***
***   LOOP. THE VALUES ARE THEN   ***
***   OUTPUT TO THE SCREEN. THIS   ***
***   ROUTINE WAS LATER USED TO    ***
***   OBTAIN SENSOR THRESHOLD VALUES ***
***   FOR USE IN THE MAIN PROGRAM  ***
***           DESIGN.              ***

```

```

100  DIM X(8)
110  M = 49280 + (4 * 16)
115  HOME
120  FOR I = 1 TO 8
130  FOR J = 1 TO 2
140  X(I) = PEEK (M + I)
150  NEXT J
160  NEXT I
200  VTAB 21: PRINT TAB( 6) "US="X(3) "
      UL="X(7)
210  VTAB 22: PRINT TAB( 3) "WS="X(2) "
      ES="X(1) " WL="X(6) " EL="X(5)
220  VTAB 23: PRINT TAB( 6) "DS="X(4) "
      DL="X(8)
400  FOR A = 1 TO 100: NEXT
410  GOTO 120

```

Figure 21. Display Routine

| | | | |
|-------|----|----|----|
| 49348 | 04 | DS | H4 |
| 49349 | 05 | EL | V1 |
| 49350 | 06 | WL | V2 |
| 49351 | 07 | UL | V3 |
| 49352 | 08 | DL | V4 |

The J Do Loop executes the PEEK command twice for each of the eight channels determined by the I loop, and assigns the proper value to the array variable X(I). Lines 200-220 provide the screen output, and line 400 is a 0.1 second delay loop added for display clarity.

D. DETERMINATION OF SENSOR VALUES

The next step in the design of the software was to use the display routine to determine the proper sensor threshold values for use in the sun tracking routine. With the routine running the sensors were exposed to a variety of conditions, and the values below were determined.

| CONDITION | VALUE | |
|---|--------|---------|
| | ANALOG | DIGITAL |
| Completely Dark | 0 V | 128 |
| Exposed to Desk Lamp | 0.08 V | 130 |
| Facing Away From Sun on Bright Day | 0.12 V | 131 |
| Directly at Bright Sun | 2.34 V | 188 |
| 4 Sensors Balanced on Bright Sun | 0.47 V | 140 |
| 4 Sensors Balanced on Hazy Sun | 0.16 V | 132 |
| 4 Sensors Balanced on Cloud Covered Sun | 0.08 V | 130 |

From the above information it was determined that a cutoff value to use for determining if the sun were out or not would be 130 or less. Limit switch values were determined to be 128 = 0V if not depressed, and 255 = +5V if depressed.

E. FLASHLIGHT FOLLOW ROUTINE

Once the drive and display routines had been developed and expected sensor values had been determined, the next step was to combine them into a working routine. The flashlight follow routine (Appendix A), was developed as an intermediate step before attacking the more complex sun tracking routine. In this routine the sensors are checked continuously to see if any go above the threshold value of 130. If any do then the appropriate drive loop is run. There are four eight step drive loops, one for each direction of travel (UP,DOWN,EAST,WEST). It should be pointed out that by running the loop in the reverse direction the motor will also run in the opposite direction. Table 5 lists the eight step loops for each direction of travel.

At the beginning of each drive loop the four A/D card channels to poke the steps to are determined by adding the appropriate channel value to the base address. For instance $L1 = M+5$ assigns the address of channel 5 to the variable L1, so that L1-L4 have the addresses of channels 5-8. The L5 and L6 variables are assigned the channel addresses of the limit switch that the system is being driven toward, and the sensor that is involved respectively. Next the limit switch

TABLE 5
EIGHT STEP DRIVE LOOPS

| | HORIZONTAL MOTOR | | VERTICAL MOTOR | |
|---------|-----------------------|------|-----------------------|------|
| | A/D CARD CHANNELS 1-4 | | A/D CARD CHANNELS 5-8 | |
| | EAST | WEST | UP | DOWN |
| STEP 1: | 1010 | 0010 | 0010 | 1010 |
| STEP 2: | 1000 | 0110 | 0110 | 1000 |
| STEP 3: | 1001 | 0100 | 0100 | 1001 |
| STEP 4: | 0001 | 0101 | 0101 | 0001 |
| STEP 5: | 0101 | 0001 | 0001 | 0101 |
| STEP 6: | 0100 | 1001 | 1001 | 0100 |
| STEP 7: | 0110 | 1000 | 1000 | 0110 |
| STEP 8: | 0010 | 1010 | 1010 | 0010 |

and sensor values are checked to be sure that both are still within limits. If not the loop is exited, and the program returns to the original sensor checking loop. If both values are still within limits the first two steps of the drive loop are sent to the appropriate drive motor. The limit switch and sensor are again checked, and either the loop is exited or two more steps are taken. This checking and stepping continues until the sensor goes below threshold (the flashlight is removed), or the system runs into the stops (the limit switch). Each of the four drive loops works the same, the only difference is the address values assigned to L1-L6.

Now that all the drive loop, sensor inputs, and limit switches had been successfully combined, the next step was to develop the sun tracking routine.

F. SUN TRACKING PROGRAM

The final step in the design of the controlling software was to develop the actual sun tracking program (Appendix B). Since the program listing is well remarked, only the main points of this program will be discussed. The program consists of four main parts. The first is a tracker calibration routine, the second is a sun location routine, the third is a sun tracking routine, and the fourth is a cloud cover waiting routine.

The calibration routine is included because the tabs that depress the various limit switches have a tendency to

become misaligned. This is important because degrees longitude, and degrees elevation are displayed on the screen. If the system is no longer limited to true east and true west or horizontal and vertical, these output values will be incorrect. When the tracker is calibrated properly it is limited to a transition of 180 degrees between the east and west limit switches, and 90 degrees between the up and down switches. Due to the gearing between the motors and drive structures of the tracker, approximately 1005 steps are required for the east to west, and 510 for the up down transitions. This routine starts the tracker in an upright east facing position, counts the steps required to transition to the west facing and down positions, and then returns to the east facing upright start position. The step count values are checked and if they are outside predetermined limits, an appropriate message is displayed on the screen. The step count values obtained are also divided by the number of degrees in their particular transition to get a steps per degree value that will be used for degrees longitude and elevation calculations in later routines.

It should be noticed that two vice four drive loops are used for the different drive requirements in the sun tracking program. As can be seen in Table 5 the east and down drive loops are the same, as are the west and up loops. Only the channel numbers of the step output lines are different. If the step and sensor/limit switch channel

addresses are determined in controlling logic external to the drive loops, and are assigned to common variables, a single drive loop can be used for two directions (EAST/DOWN or WEST/UP).

The sun location routine is used to obtain a rough lock on before passing control to the sun tracking routine. Starting from the east facing upright position, the system is moved toward the west. The up sensor values are checked and if they remain the same or increase the sensor is going toward the sun. When the values begin to decrease a counter is set and if the old sensor reading is less than the new value three times in a row, the tracker moves back to the position of the first decrease. If the readings are the same the counter stays the same, if the new reading is more than the last the counter is reset. These counters are necessary because the sensor values tend to vary by ± 1 quite often. If there are three successive decreases then random value fluctuations can be ruled out.

When the tracker has moved back to the point of maximum sun intensity (the first sensor value decrease), it begins an upward transition. During this transition the up and down values are continually checked. When their values are the same the sun's intensity is equal on both sensors, thus the tracker is locked onto the sun's elevation. Control is now passed to the sun tracking routine. If lock-on is not

achieved in either the horizontal or the vertical transitions, an appropriate message is printed on the screen.

The tracking routine's main function is to ensure that the UP/DOWN sensor values are the same, and that the EAST/WEST sensors are also balanced. The four values do not necessarily have to be the same, only each pair must balance. When this is achieved then the tracker is locked onto the sun. When one sensor of a pair is not equal to the other a counter is set, if the sensors remain unbalanced for a count of five, two drive steps are taken in the direction necessary to rebalance the sensor pair. Again the counter is necessary due to the sensor outputs tendency to vary about a value by ± 1 . If clouds cover the sun, its intensity becomes too low for the sensors to track, and control is passed to the cloud cover routine.

The cloud cover routine continually checks the up sensor to determine if it goes back above the threshold value (clouds no longer cover the sun). If it does, control is returned to the tracking routine. If the value remains below threshold a counter is incremented once each second. At 10 minutes or a counter value of 600, control is passed back to the sun location routine, which places the tracker back in startup position and waits for the sun to reappear. The 10 minute limit is necessary because after prolonged cloud cover, the sun will move out of position and relocation will be necessary.

As mentioned the program listing (Appendix B), is well remarked and can be traced through for program specifics.

IV. CONCLUSIONS AND RECOMMENDATIONS

Once completed the system performed well with the new computer. The screen/keyboard interaction now available expands the possible system uses. Some possible system enhancements can be made.

A clock card could be added to the Apple that would bring a real time capability to the system. By being able to determine the date and time, a series of calculations could be performed that would give the sun's location. Reference 1 in its chapter on "Terrestrial Solar Radiation," describes the procedure in some detail. Once latitude and elevation values had been calculated for the sun, the tracker could be placed in that position and control passed to the sun tracking routine. This would do away with the need for the sun location routine, and make the initial lock-on much faster. Another use would be in the cloud cover routine where the tracker's position could be updated by the same calculations. Since the tracker's position could now be automatically updated, the system would not have to return to the start up position after the 10 minute time out.

Another enhancement would be to assemble the drive routines in order to increase drive speed. The assembled code would be internal to the system thus preserving the easier to interpret Basic code for user understanding. The

steps could be delivered to the motors much more rapidly than in the Basic form, thus preventing the instability problem encountered earlier with the four step drive loop.

Another idea would be to use the tracker as a positioning device for a larger tracking system. The longitude and elevation position information that is now displayed on the screen could also be sent to a much larger device, thus eliminating the need for sensors on that device. Drive motor commands could also be sent to power amplification devices that would drive the larger positioning motors.

A card could be developed and built to be placed inside the Apple computer that would replace the circuit board currently inside the A/I box. This would reduce the need for much of the current external wiring, as the signals would interface the computer via the internal bus. With the circuit board out of the A/I box the power transistors could be relocated into a much smaller box which would be mounted under the tracker. This would produce a much more compact system.

One last recommendation is to develop a power supply to deliver all the necessary voltages to the system. This would do away with the current requirement for three separate external power supplies.

APPENDIX A

FLASHLIGHT FOLLOW ROUTINE

```

10  REM  "MAIN ROUTINE"
20  REM
30  REM  "FLASHLIGHT FOLLOW"
40  REM  "ROUTINE"
50  REM
60  REM  "SET HIGH AND LOW"
70  REM  "MOTOR DRIVE VALUES"
80  REM  "FOR DRIVE ROUTINES"
90  REM
100 HI = 255
110 LO = 128
120 REM
130 REM  "SET LIMIT SWITCH AND"
140 REM  "SENSOR THRESHOLDS"
150 REM
160 THR = 130
170 CO = 160
180 REM
190 REM  "DETERMINE MEMORY"
200 REM  "START ADDRESS FOR"
210 REM  "PROGRAM USE"
220 REM
230 M = 49280 + (4 * 16)
240 REM
250 REM  "CHECK SENSOR VALUES"
260 REM  "TO DETERMINE IF ANY"
270 REM  "ARE ABOVE THRESHOLD"
280 REM
290 FOR I = 1 TO 4
300 FOR J = 1 TO 2
310 X(I) = PEEK (M + I).
320 NEXT J
330 NEXT I
340 REM
350 REM  "IF ANY ABOVE"
360 REM  "THRESHOLD GO TO"
370 REM  "APPROPRIATE DRIVE"
380 REM  "SUBROUTINE"
390 REM
400 IF X(3) > = THR THEN GOTO 2000
410 IF X(4) > = THR THEN GOTO 3000
420 IF X(1) > = THR THEN GOTO 4000
430 IF X(2) > = THR THEN GOTO 5000
440 GOTO 290
450 REM

```

```

2000 REM "UP DRIVE SUBROUTINE"
2010 REM
2020 REM "INITIALIZE VALUES"
2030 REM "TO BE USED IN"
2040 REM "SUBROUTINE"
2050 REM
2060 L1 = M + 5:
      L2 = M + 6:
      L3 = M + 7:
      L4 = M + 8:
      L5 = M + 7:
      L6 = M + 3
2070 REM
2080 REM "DETERMINE IF ARM"
2090 REM "IS AGAINST UPPER"
2100 REM "LIMIT SWITCH OR IF"
2110 REM "UPPER SENSOR IS"
2120 REM "STILL ABOVE THRES-"
2130 REM "HOLD VALUE"
2140 REM
2150 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
2160 REM
2170 REM "IF SWITCH DEPRESSED"
2180 REM "OR IF SENSOR BELOW"
2190 REM "THRESHOLD RETURN TO"
2200 REM "MAIN LOOP"
2210 REM
2220 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
2230 REM
2240 REM "PLUG IN (POKE)"
2250 REM "VALUES FOR FIRST"
2260 REM "TWO 0.9 DEG. STEPS"
2270 REM
2280 POKE L1,L0:
      POKE L2,L0:
      POKE L3,HI:
      POKE L4,L0
2290 POKE L1,L0:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,L0
2300 REM
2310 REM "RECHECK LIMIT"
2320 REM "SWITCH AND SENSOR"
2330 REM "VALUES"
2340 REM
2350 Y1 = PEEK (L5):
      Y1 = PEEK (L5):

```

```

        Y2 = PEEK (L6):
        Y2 = PEEK (L6)
2360  IF (Y1 > = CO) OR (Y2 < = THR) THEN  GOTO 280
2370  REM
2380  REM  "MOTOR SETPS THREE"
2390  REM  "AND FOUR"
2400  REM
2410  POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,LO
2420  POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
2430  REM
2440  REM  "RECHECK THRESHOLD"
2450  REM  "VALUES"
2460  REM
2470  Y1 = PEEK (L5):
        Y1 = PEEK (L5):
        Y2 = PEEK (L6):
        Y2 = PEEK (L6)
2480  IF (Y1 > = CO) OR (Y2 < = THR) THEN  GOTO 280
2490  REM
2500  REM  "MOTOR STEPS FIVE"
2510  REM  "AND SIX"
2520  REM
2530  POKE L1,LO:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
2540  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
2550  REM
2560  REM  "RECHECK THRESHOLD"
2570  REM  "VALUES"
2580  REM
2590  Y1 = PEEK (L5):
        Y1 = PEEK (L5):
        Y2 = PEEK (L6):
        Y2 = PEEK (L6)
2600  IF (Y1 > = CO) OR (Y2 < = THR) THEN  GOTO 280
2610  REM
2620  REM  "MOTOR STEPS SEVEN"
2630  REM  "AND EIGHT"
2640  REM
2650  POKE L1,HI:
        POKE L2,LO:

```

```

        POKE L3,L0:
        POKE L4,L0
2660    POKE L1,HI:
        POKE L2,L0:
        POKE L3,HI:
        POKE L4,L0
2670    GOTO 2060
3000    REM  "DOWN DRIVE ROUTINE"
3010    REM
3020    REM  "INITIALIZE  VALUES"
3030    REM  "TO BE USED IN"
3040    REM  "SUBROUTINE"
3050    REM
3060    L1 = M + 5:
        L2 = M + 6:
        L3 = M + 7:
        L4 = M + 8:
        L5 = M + 8:
        L6 = M + 4
3070    REM
3080    REM  "DETERMINE IF ARM"
3090    REM  "IS AGAINST DOWN"
3100    REM  "LIMIT SWITCH OR IF"
3110    REM  "DOWN SENSOR IS"
3120    REM  "STILL ABOVE THRES-"
3130    REM  "HOLD VALUE"
3140    REM
3150    Y1 = PEEK (L5):
        Y1 = PEEK (L5):
        Y2 = PEEK (L6):
        Y2 = PEEK (L6)
3160    REM
3170    REM  "IF SWITCH DEPRESSED"
3180    REM  "OR IF SENSOR BELOW"
3190    REM  "THRESHOLD RETURN TO"
3200    REM  "MAIN LOOP"
3210    REM
3220    IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
3230    REM
3240    REM  "PLUG IN (POKE)"
3250    REM  "VALUES FOR FIRST"
3260    REM  "TWO 0.9 DEG. STEPS"
3270    REM
3280    POKE L1,HI:
        POKE L2,L0:
        POKE L3,HI:
        POKE L4,L0
3290    POKE L1,HI:
        POKE L2,L0:
        POKE L3,L0:
        POKE L4,L0

```

```

3300 REM
3310 REM "RECHECK LIMIT"
3320 REM "SWITCH AND SENSOR"
3330 REM "VALUES"
3340 REM
3350 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
3360 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
3370 REM
3380 REM "MOTOR STEPS THREE"
3390 REM "AND FOUR"
3400 REM
3410 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
3420 POKE L1,LO:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
3430 REM
3440 REM "RECHECK THRESHOLD"
3450 REM "VALUES"
3460 REM
3470 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
3480 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
3490 REM
3500 REM "MOTOR STEPS FIVE"
3510 REM "AND SIX"
3520 REM
3530 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,HI
3540 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,LO
3550 REM
3560 REM "RECHECK THRESHOLD"
3570 REM "VALUES"
3580 REM
3590 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)

```



```

3600 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
3610 REM
3620 REM "MOTOR STEPS SEVEN"
3630 REM "AND EIGHT"
3640 REM
3650 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
3660 POKE L1,LO:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
3670 GOTO 3060
4000 REM "EAST DRIVE ROUTINE"
4010 REM
4020 REM "INITIALIZE VALUES"
4030 REM "TO BE USED IN"
4040 REM "SUBROUTINE"
4050 REM
4060 L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = M + 5:
      L6 = M + 1
4070 REM
4080 REM "DETERMINE IF ARM"
4090 REM "IS AGAINST EAST"
4100 REM "LIMIT SWITCH OR IF"
4110 REM "EAST SENSOR IS"
4120 REM "STILL ABOVE THRES-"
4130 REM "HOLD VALUE"
4140 REM
4150 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
4160 REM
4170 REM "IF SWITCH DEPRESSED"
4180 REM "OR IF SENSOR BELOW"
4190 REM "THRESHOLD RETURN TO"
4200 REM "MAIN LOOP"
4210 REM
4220 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
4230 REM
4240 REM "PLUG IN (POKE)"
4250 REM "VALUES FOR FIRST"
4260 REM "TWO 0.9 DEG. STEPS"
4270 REM
4280 POKE L1,HI:

```

```

        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
4290    POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
4300    REM
4310    REM  "RECHECK LIMIT"
4320    REM  "SWITCH AND SENSOR"
4330    REM  "VALUES"
4340    REM
4350    Y1 =  PEEK (L5):
        Y1 =  PEEK (L5):
        Y2 =  PEEK (L6):
        Y2 =  PEEK (L6)
4360    IF (Y1 > = CO) OR (Y2 < = THR) THEN  GOTO 280
4370    REM
4380    REM  "MOTOR STEPS THREE"
4390    REM  "AND FOUR"
4400    REM
4410    POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
4420    POKE L1,LO:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
4430    REM
4440    REM  "RECHECK THRESHOLD"
4450    REM  "VALUES"
4460    REM
4470    Y1 =  PEEK (L5):
        Y1 =  PEEK (L5):
        Y2 =  PEEK (L6):
        Y2 =  PEEK (L6)
4480    IF (Y1 > = CO) OR (Y2 < = THR) THEN  GOTO 280
4490    REM
4500    REM  "MOTOR STEPS FIVE"
4510    REM  "AND SIX"
4520    REM
4530    POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
4540    POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,LO
4550    REM

```

```

4560 REM "RECHECK THRESHOLD"
4570 REM "VALUES"
4580 REM
4590 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
4600 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
4610 REM
4620 REM "MOTOR STEPS SEVEN"
4630 REM "AND EIGHT"
4640 REM
4650 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
4660 POKE L1,LO:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
4670 GOTO 4060
5000 REM "WEST DRIVE ROUTINE"
5010 REM
5020 REM "INITIALIZE VALUES"
5030 REM "TO BE USED IN"
5040 REM "SUBROUTINE"
5050 REM
5060 L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = M + 6:
      L6 = M + 2
5070 REM
5080 REM "DETERMINE IF ARM"
5090 REM "IS AGAINST WEST"
5100 REM "LIMIT SWITCH OR IF"
5110 REM "WEST SENSOR IS"
5120 REM "STILL ABOVE THRES-"
5130 REM "HOLD VALUE"
5140 REM
5150 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
5160 REM
5170 REM "IF SWITCH DEPRESSED"
5180 REM "OR IF SENSOR BELOW"
5190 REM "THRESHOLD RETURN TO"
5200 REM "MAIN LOOP"
5210 REM

```

```

5220 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
5230 REM
5240 REM "PLUG IN (POKE)"
5250 REM "VALUES FOR FIRST"
5260 REM "TWO 0.9 DEG. STEPS"
5270 REM
5280 POKE L1,L0:
      POKE L2,L0:
      POKE L3,HI:
      POKE L4,L0
5290 POKE L1,L0:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,L0
5300 REM
5310 REM "RECHECK LIMIT"
5320 REM "SWITCH AND SENSOR"
5330 REM "VALUES"
5340 REM
5350 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
5360 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
5370 REM
5380 REM "MOTOR SETPS THREE"
5390 REM "AND FOUR"
5400 REM
5410 POKE L1,L0:
      POKE L2,HI:
      POKE L3,L0:
      POKE L4,L0
5420 POKE L1,L0:
      POKE L2,HI:
      POKE L3,L0:
      POKE L4,HI
5430 REM
5440 REM "RECHECK THRESHOLD"
5450 REM "VALUES"
5460 REM
5470 Y1 = PEEK (L5):
      Y1 = PEEK (L5):
      Y2 = PEEK (L6):
      Y2 = PEEK (L6)
5480 IF (Y1 > = CO) OR (Y2 < = THR) THEN GOTO 280
5490 REM
5500 REM "MOTOR STEPS FIVE"
5510 REM "AND SIX"
5520 REM
5530 POKE L1,L0:
      POKE L2,L0:

```

```

        POKE L3,LO:
        POKE L4,HI
5540    POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
5550    REM
5560    REM  "RECHECK THRESHOLD"
5570    REM  "VALUES"
5580    REM
5590    Y1 =  PEEK (L5):
        Y1 =  PEEK (L5):
        Y2 =  PEEK (L6):
        Y2 =  PEEK (L6)
5600    IF (Y1 >  = CO) OR (Y2 <  = THR) THEN  GOTO 280
5610    REM
5620    REM  "MOTOR STEPS SEVEN"
5630    REM  "AND EIGHT"
5640    REM
5650    POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
5660    POKE L1,HI:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
5670    GOTO 5060

```

APPENDIX B

MAIN SUN TRACKING PROGRAM

```

100 REM *****
110 REM ** THE FOLLOWING IS A PROGRAM FOR SUN **
120 REM ** TRACKING. IT CONSISTS OF FOUR MAIN **
130 REM ** PARTS. (1) A SYSTEM CALIBRATION **
140 REM ** ROUTINE (2) A SUN LOCATION ROUTINE **
150 REM ** (3) A SUN TRACKING ROUTINE (4) AND **
160 REM ** A CLOUD COVER WAIT ROUTINE. **
170 REM *****
180 REM
190 REM
200 REM *****
210 REM *** CALIBRATION ROUTINE ***
220 REM *****
230 REM
240 REM
250 REM *** CLEAR SCREEN ***
260 HOME
270 REM
280 REM *** PRINT HEADER ON SCREEN ***
290 REM
300 VTAB 2: PRINT "THE NUMBER OF 0.9 DEG. STEPS REQUIRED"
310 VTAB 3: PRINT "TO DRIVE THE SYSTEM THROUGH A COMPLETE"
320 VTAB 4: PRINT "HORIZONTAL OR VERTICAL TRANSITION ARE"
330 VTAB 5: PRINT "COUNTED AND USED TO DETERMINE TRACKING"
340 VTAB 6: PRINT "POSITION, AND CHECK SYSTEM CALIBRATION"
341 REM
342 REM *** DUE TO GEARING DOWN OF ***
343 REM *** THE DRIVE MOTORS APPROX ***
344 REM *** 500 0.9DEG STEPS ARE ***
345 REM *** REQUIRED TO TRANSITION 90 ***
346 REM *** DEGREES. ***
347 REM
350 REM
360 REM *** CALCULATE MEMORY LOCATION ***
370 REM *** POINTER FOR A/D CARD. ***
380 REM
390 M = 49280 + (4 * 16)
400 REM
410 REM *** SET VALUE TO DETERMINE IF ***
420 REM *** LIMIT SWITCH IS DEPRESSED ***
430 REM
440 CO.= 160
450 REM

```



```

460 REM *** SET VALUES FOR STEP POKES ***
470 REM *** TO A/D CARD,127=0V 255=5V ***
480 REM
490 HI = 255
500 LO = 127
510 REM
520 REM *** INITIALIZE COUNTERS TO BE ***
530 REM *** USED IN SUN LOCATION RTN. ***
540 REM
550 AB = 0:AC = 0
560 REM
570 REM *** THESE TWO CALLS INITIALIZE***
580 REM *** THE SEEKER BY PLACING IT ***
590 REM *** IN AN UPRIGHT EAST FACING ***
600 REM *** POSITION ***
610 REM
620 VTAB 8: PRINT "PLACE IN UPRIGHT POSITION"
630 REM
640 REM *** PASS TO DRIVE ROUTINES THE***
650 REM *** MEMORY LOCATIONS OF THE ***
660 REM *** A/D CARD TO POKE STEP ***
670 REM *** VALUES TO (L1-L4). ALSO ***
680 REM *** MEMORY LOCATION OF LIMIT ***
690 REM *** SWITCH TO CHECK FOR ***
700 REM *** DEPRESSION AT END OF RUN ***
710 REM *** (L5), AND PASS VALUE TO ***
720 REM *** TO TELL ROUTINE IF IT IS ***
730 REM *** DRIVING UP OR EAST (L6) ***
740 REM
750 L1 = M + 5:
    L2 = M + 6:
    L3 = M + 7:
    L4 = M + 8:
    L5 = M + 8:
    L6 = 1
760 GOTO 1810
770 REM
780 REM *** DRIVE TO EAST. L1-L6 SAME ***
790 REM *** USE AS ABOVE. ***
800 REM
810 VTAB 8: PRINT "SET FACING EAST"
820 L1 = M + 1:
    L2 = M + 2:
    L3 = M + 3:
    L4 = M + 4:
    L5 = M + 5:
    L6 = 2
830 GOTO 1810
840 REM
850 REM *** THESE TWO CALLS CAUSE THE ***
860 REM *** NUMBER OF STEPS FOR ONE ***

```

```

870 REM *** TRANSITION WEST AND ONE ***
880 REM *** TRANSITION TO THE DOWN ***
890 REM *** POSITION TO BE COUNTED ***
900 REM
910 VTAB 8: PRINT "WEST COUNTING RUN"
920 REM
930 REM *** EAST COUNTING RUN. PASS ***
940 REM *** TO DRIVE ROUTINE STEP ***
950 REM *** POKE MEMORY LOCATIONS ***
960 REM *** (L1-L4), LIMIT SWITCH (L5) ***
970 REM *** WHERE TO RETURN WHEN ***
980 REM *** FINISH COUNTING RUN (L6) ***
990 REM *** AND INDICATOR OF WEST OR ***
1000 REM *** UP COUNTING RUN (L7), ***
1010 REM
1020 L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = M + 6:
      L6 = 1:
      L7 = 200
1030 GOTO 2320
1040 REM
1050 REM *** PRINT OUT NUMBER OF STEPS ***
1060 REM *** FOR EAST TO WEST TRANSI- ***
1070 REM *** TION (LN), AND STEPS PER ***
1080 REM *** DEGREE (LD). ***
1090 REM
1100 VTAB 14: PRINT "LONGITUDE COUNT="LN
1110 LD = LN / 180
1120 REM
1130 REM *** LIMIT SCREEN DISPLAY TO ***
1140 REM *** TWO DECIMAL POINT ACCURACY ***
1150 REM
1160 LD = INT (LD * 100 + .5) / 100
1170 VTAB 18: PRINT "STEPS/DEG. LONG="LD
1180 REM
1190 REM *** COUNTING RUN FROM UPRIGHT ***
1200 REM *** TO DOWN POSITION ***
1210 REM
1220 VTAB 8: PRINT "VERT COUNTING RUN"
1230 REM
1240 REM *** PASS SAME INFORMATION AS ***
1250 REM *** AS ABOVE TO DRIVE ROUTINE ***
1260 REM
1270 L1 = M + 5:
      L2 = M + 6:
      L3 = M + 7:
      L4 = M + 8:
      L5 = M + 7:

```

```

      L6 = 2:
      L7 = 155
1280  GOTO 2320
1290  REM
1300  REM *** PRINTOUT SAME AS ABOVE      ***
1310  REM *** FOR STEPS FROM UP TO DOWN ***
1320  REM *** POSITION (LT), AND STEPS    ***
1330  REM *** PER DEGREE (LE).          ***
1340  REM
1350  VTAB 15: PRINT "LATITUDE COUNT ="LT
1360  LE = LT / 90
1370  LE = INT (LE * 100 + .5) / 100
1380  VTAB 19: PRINT "STEPS/DEG. LAT ="LE
1390  REM
1400  REM *** THESE TWO CALLS RETURN      ***
1410  REM *** THE SEEKER TO THE UPRIGHT ***
1420  REM *** AND EAST FACING POSITION    ***
1430  REM *** BEFORE GOING TO THE SUN   ***
1440  REM *** LOCATION ROUTINE          ***
1450  REM
1460  VTAB 8: PRINT "RETURN TO START POSITION"
1470  L1 = M + 5:
      L2 = M + 6:
      L3 = M + 7:
      L4 = M + 8:
      L5 = M + 8:
      L6 = 3
1480  GOTO 1810
1490  L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = M + 5:
      L6 = 4
1500  GOTO 1810
1510  HOME
1520  REM
1530  REM *** DETERMINE IF THE NUMBER    ***
1540  REM *** OF STEPS FOR EACH TRANSI- ***
1550  REM *** TION IS OUT OF LIMITS, IF ***
1560  REM *** IT IS PRINT APPROPRIATE   ***
1570  REM *** ERROR MESSAGE.            ***
1580  REM
1590  IF (LN < 990) OR (LN > 1020) THEN VTAB 10:
      PRINT "HORIZONTAL PARAMETERS OUT OF RANGE": VTAB 11:
      PRINT "CHECK LIMIT SWITCH ALLIGNMENT": GOTO 1660
1600  IF (LT < 490) OR (LT > 520) THEN VTAB 12:
      PRINT "VERTICAL PARAMETERS OUT OF RANGE": VTAB 13:
      PRINT "CHECK LIMIT SWITCH ALLIGNMENT": GOTO 1660
1610  GOTO 5090
1620  REM

```

```

1630 REM *** DETERMINE IF WANT TO ***
1640 REM *** CONTINUE WHEN OUT OF LIMIT***
1650 REM
1660 VTAB 15: INPUT "DO YOU WANT TO CONTINUE?(Y/N)";K$
1670 IF K$ = "N" THEN PRINT CHR$ (4);"RUN HELLO"
1680 IF K$ = "Y" THEN GOTO 5090
1690 REM
1700 REM *** DRIVE ROUTINE FOR EAST AND***
1710 REM *** UP INITIALIZATION. ***
1720 REM
1730 REM
1740 REM
1750 REM *** CHECK LIMIT SWITCH AS ***
1760 REM *** DETERMINED BY L5, TO ***
1770 REM *** SEE IF SEEKER ARM IS ***
1780 REM *** AGAINST IT. IF IT IS JUMP ***
1790 REM *** OUT OF THE ROUTINE. ***
1800 REM
1810 Y1 = PEEK (L5):Y1 = PEEK (L5)
1820 IF Y1 > C0 THEN GOTO 2220
1830 REM
1840 REM *** VALUES TO POKE INTO D/A ***
1850 REM *** CARD MEMORY LOCATIONS ***
1860 REM *** (L1-L4) TO PRODUCE MOTOR ***
1870 REM *** STEP. REPEAT FOR 8 STEP ***
1880 REM *** CONTINUOUS LOOP. ***
1890 REM
1900 POKE L1,HI:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
1910 Y1 = PEEK (L5):
      Y1 = PEEK (L5)
1920 IF Y1 > C0 THEN GOTO 2220
1930 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,LO
1940 Y1 = PEEK (L5):
      Y1 = PEEK (L5)
1950 IF Y1 > C0 THEN GOTO 2220
1960 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
1970 Y1 = PEEK (L5):
      Y1 = PEEK (L5)
1980 IF Y1 > C0 THEN GOTO 2220
1990 POKE L1,LO:
      POKE L2,LO:
      POKE L3,LO:

```

```

        POKE L4,HI
2000    Y1 = PEEK (L5):
        Y1 = PEEK (L5)
2010    IF Y1 > C0 THEN GOTO 2220
2020    POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
2030    Y1 = PEEK (L5):
        Y1 = PEEK (L5)
2040    IF Y1 > C0 THEN GOTO 2220
2050    POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,LO
2060    Y1 = PEEK (L5):
        Y1 = PEEK (L5)
2070    IF Y1 > C0 THEN GOTO 2220
2080    POKE L1,LO:
        POKE L2,HI:
        POKE L3,HI:
        POKE L4,LO
2090    Y1 = PEEK (L5):
        Y1 = PEEK (L5)
2100    IF Y1 > C0 THEN GOTO 2220
2110    POKE L1,LO:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
2120    REM
2130    REM *** RETURN TO STEP ONE          ***
2140    REM
2150    GOTO 1810
2160    REM
2170    REM *** RETURN TO THE CORECT      ***
2180    REM *** PORTION OF THE CONTROL    ***
2190    REM *** SECTION, AS DETERMINED   ***
2200    REM *** BY L6                     ***
2210    REM
2220    ON L6 GOTO 810,910,1490,1510
2230    REM
2240    REM
2250    REM *** WEST AND DOWN COUNTING    ***
2260    REM *** DRIVE ROUTINE.           ***
2270    REM
2280    REM
2290    REM
2300    REM *** INITIALIZE COUNTER (N)    ***
2310    REM
2320    N = 0
2330    REM

```



```

2340 REM *** STEP ONE OF EIGHT      ***
2350 REM
2360 POKE L1,L0:
      POKE L2,L0:
      POKE L3,HI:
      POKE L4,L0
2370 REM
2380 REM *** INCRIMENT COUNTER      ***
2390 REM
2400 N = N + 1
2410 REM
2420 REM *** PRINT COUNT VALUE AND   ***
2430 REM *** ASSIGN IT TO CORRECT    ***
2440 REM *** VARIABLE AS DETERMINED  ***
2450 REM *** BY L7                    ***
2460 REM
2470 IF L7 = 200 THEN VTAB 10:
      PRINT "HORIZCT="N:LN = N
2480 IF L7 = 155 THEN VTAB 12:
      PRINT "VERTCT ="N:LT = N
2490 POKE L1,L0:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,L0
2500 REM
2510 REM *** CHECK THE LIMIT SWITCH  ***
2520 REM *** ASSIGNED BY L5. IF THE  ***
2530 REM *** SEEKER IS AGAINST IT EXIT ***
2540 REM *** THE DRIVE ROUTINE.      ***
2550 REM *** NOTE:THE TASKS ARE DIVIDED***
2560 REM *** AND ASSIGNED TO EVERY   ***
2570 REM *** OTHER STEP FOR DRIVE SPEED***
2580 REM
2590 Y1 = PEEK (L5)
2600 Y1 = PEEK (L5)
2610 IF Y1 > = C0 THEN GOTO 3050
2620 N = N + 1
2630 REM
2640 REM *** CONTINUE FOR EIGHT STEP  ***
2650 REM *** LOOP.                    ***
2660 REM
2670 POKE L1,L0:
      POKE L2,HI:
      POKE L3,L0:
      POKE L4,L0
2680 N = N + 1
2690 IF L7 = 200 THEN VTAB 10:
      PRINT "HORIZCT="N:LN = N
2700 IF L7 = 155 THEN VTAB 12:
      PRINT "VERTCT ="N:LT = N
2710 POKE L1,L0:

```



```

        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
2720  Y1 = PEEK (L5)
2730  Y1 = PEEK (L5)
2740  IF Y1 > = CO THEN GOTO 3050
2750  N = N + 1
2760  POKE L1,LO:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
2770  N = N + 1
2780  IF L7 = 200 THEN VTAB 10:
        PRINT "HORIZCT="N:LN = N
2790  IF L7 = 155 THEN VTAB 12:
        PRINT "VERTCT ="N:LT = N
2800  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
2810  Y1 = PEEK (L5)
2820  Y1 = PEEK (L5)
2830  IF Y1 > = CO THEN GOTO 3050
2840  N = N + 1
2850  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
2860  N = N + 1
2870  IF L7 = 200 THEN VTAB 10:
        PRINT "HORIZCT="N:LN = N
2880  IF L7 = 155 THEN VTAB 12:
        PRINT "VERTCT ="N:LT = N
2890  POKE L1,HI:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
2900  Y1 = PEEK (L5)
2910  Y1 = PEEK (L5)
2920  IF Y1 > = CO THEN GOTO 3050
2930  N = N + 1
2940  REM
2950  REM *** RETURN TO STEP ONE          ***
2960  REM
2970  GOTO 2360
2980  REM
2990  REM *** WHEN WEST COUNTING RUN      ***
3000  REM *** COMPLETE SWITCH TO DOWN    ***
3010  REM *** COUNTING RUN. WHEN COUNT    ***
3020  REM *** RUNS COMPLETE RETURN TO    ***
3030  REM *** EAST/UP START POSITION      ***

```

```

3040 REM
3050 ON L6 GOTO 1100,1350

5000 REM
5005 REM
5010 REM *****
5015 REM **          SUN LOCATION ROUTINE          **
5020 REM *****
5025 REM
5030 REM
5035 REM *** THIS ROUTINE LOCATES THE ***
5040 REM *** SUN IN A GENERAL AREA AND ***
5045 REM *** PASSES CONTROL TO THE SUN ***
5050 REM *** TRACKING ROUTINE FOR FINE ***
5055 REM *** ADJUSTING AND LOCK ON ***
5060 REM *** FOLLOWING. ***
5065 REM
5070 REM
5075 REM
5080 REM *** INITIALIZE COUNTERS ***
5085 REM
5090 DH = 0
5095 Z = 0
5100 G = 0
5105 C2 = 0
5110 HOME
5115 REM
5120 REM *** IF FAIL TO LOCATE THE SUN ***
5125 REM *** AFTER TWO TRYs GO TO ***
5130 REM *** ERROR MESSAGE. ***
5135 REM
5140 IF AB = 2 GOTO 6025
5145 AB = AB + 1
5150 VTAB 2: PRINT "THIS ROUTINE FINDS THE SUN'S POSITION"
5155 VTAB 3: PRINT "AND PASSES CONTROL TO SUN TRACKING"
5160 VTAB 4: PRINT "ROUTINES"
5165 REM
5168 REM *** A/D CARD MEMORY LOCA- ***
5170 REM *** TIONS FOR DRIVE STEPS ***
5175 REM
5180 L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4

5185 REM
5190 REM *** CHECK UPSENSOR VALUE IF ***
5195 REM *** GREATER THAN CUTOFF VALUE ***
5200 REM *** GO TO SUN LOCATION LOGIC ***
5205 REM *** IF NOT CONTINUE TO LOOP ***
5210 REM
5215 C2 = PEEK (M + 3):

```

```

      C2 = PEEK (M + 3)
5220 IF C2 > = 130 THEN GOTO 5250
5225 VTAB 6: PRINT "US="C2
5230 GOTO 5215
5235 REM
5240 REM ***      SUN LOCATION LOGIC      ***
5245 REM
5250 VTAB 9: PRINT "EAST=0 DEG:SOUTH=90 DEG:WEST=180 DEG
5255 REM
5260 REM *** FIRST STEP                      ***
5265 REM
5270 POKE L1,LO:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
5275 REM
5280 REM *** CHECK UPSENSOR (C2), AND      ***
5285 REM *** WEST LIMIT SWITCH (LW),      ***
5290 REM *** VALUES.                    ***
5295 REM
5300 C2 = PEEK (M + 3):
      C2 = PEEK (M + 3)
5305 LW = PEEK (M + 6):
      LW = PEEK (M + 6)
5310 REM
5315 REM *** PRINT UPSENSOR VALUE ON      ***
5320 REM *** SCREEN, ENSURE SUN BRIGHT   ***
5325 REM *** ENOUGH (C2>131), THAT ARM   ***
5330 REM *** IS NOT AGAINST WEST LIMIT    ***
5335 REM *** SWITCH (LW<160), AND        ***
5340 REM *** INCRIMENT DEGREES HORIZ.    ***
5345 REM *** COUNTER (DH)                 ***
5350 REM
5355 VTAB 6: PRINT "UPSENSOR READING ="C2
5360 IF C2 < 131 THEN GOTO 5945
5365 IF LW > 160 THEN GOTO 5985
5370 DH = DH + 1
5375 REM
5380 REM *** SECOND STEP                      ***
5385 REM
5390 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
5395 REM
5400 REM *** IF OF UPSENSOR OBTAINED      ***
5405 REM *** THIS TIME (C2) IS GREATER    ***
5410 REM *** THAN THE VALUE OBTAINED      ***
5415 REM *** LAST TIME (C1), ZERO THE     ***
5420 REM *** COUNTER BECAUSE STILL        ***
5425 REM *** HEADING TOWARD THE SUN      ***

```

```

5430 REM
5435 IF C2 > C1 THEN Z = 0:G = 0
5440 REM
5445 REM *** IF CURRENT VALUE LESS ***
5450 REM *** THAN LAST VALUE INCRIMENT ***
5455 REM *** COUNTER (Z), AND STEPS ***
5460 REM *** TAKEN SINCE MAX VALUE ***
5465 REM *** COUNTER (G). ***
5470 REM
5475 IF C2 < C1 THEN Z = Z + 1:G = G + 2
5480 REM
5485 REM *** CONTINUE COUNTING STEPS ***
5490 REM *** SINCE PASSED MAX VALUE ***
5495 REM *** OF UPSENSOR. ***
5500 REM
5505 IF (C2 = C1) AND (Z > 0) THEN G = G + 2
5510 REM
5515 REM *** REPLACE PREVIOUS READING ***
5520 REM *** WITH CURRENT READING FOR ***
5525 REM *** NEXT CHECK. ***
5530 REM
5535 C1 = C2
5540 REM
5545 REM *** IF CURRENT READING WAS ***
5550 REM *** LESS THAN PREVIOUS READING***
5555 REM *** FOR THREE CONSECUTIVE ***
5560 REM *** CHECKS, THEN ARE GOING ***
5565 REM *** AWAY FROM SUN. GO BACK TO ***
5570 REM *** WHERE FIRST C2<C1 OCCURED ***
5575 REM *** (G) STEPS AGO. ***
5580 REM
5585 IF Z = 3 THEN AB = 0: GOTO 6090
5590 REM
5595 REM *** INCRIMENT DEGREES HORIZ. ***
5600 REM *** COUNTER (DH). REPEAT ABOVE***
5605 REM *** LOGIC ON ALTERNATE STEP ***
5610 REM *** BASIS FOR REMAINING STEPS ***
5615 REM *** OF EIGHT STEP DRIVE LOOP ***
5620 REM
5625 DH = DH + 1
5630 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,LO
5635 C2 = PEEK (M + 3):
      C2 = PEEK (M + 3)
5640 LW = PEEK (M + 6):
      LW = PEEK (M + 6)
5645 VTAB 6: PRINT "UPSENSOR READING ="C2
5650 IF C2 < 131 THEN GOTO 5945
5655 IF LW > 160 THEN GOTO 5985

```

```

5660 X1 = DH / LD
5665 DH = DH + 1
5670 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,HI
5675 IF C2 > C1 THEN Z = 0:G = 0
5680 IF C2 < C1 THEN Z = Z + 1:G = G + 2
5685 IF (C2 = C1) AND (Z > 0) THEN G = G + 2
5690 C1 = C2
5695 IF Z = 3 THEN AB = 0: GOTO 6090
5700 DH = DH + 1
5705 REM
5710 REM *** SET UP FOR DEGREES LONG. ***
5715 REM *** PRINTOUT WITH ONE DECIMAL ***
5720 REM *** PLACE ACCURACY. ***
5725 REM
5730 X1 = INT (X1 * 10 + .5) / 10
5735 POKE L1,LO:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
5740 C2 = PEEK (M + 3):
      C2 = PEEK (M + 3)
5745 LW = PEEK (M + 6):
      LW = PEEK (M + 6)
5750 VTAB 6: PRINT "UPSENSOR READING ="C2
5755 IF C2 < 131 THEN GOTO 5945
5760 IF LW > 160 THEN GOTO 5985
5765 VTAB 8: PRINT "DEGREES LONGITUDE ="X1
5770 DH = DH + 1
5775 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
5780 IF C2 > C1 THEN Z = 0:G = 0
5785 IF C2 < C1 THEN Z = Z + 1:G = G + 2
5790 IF (C2 = C1) AND (Z > 0) THEN G = G + 2
5795 C1 = C2
5800 IF Z = 3 THEN AB = 0: GOTO 6090
5805 DH = DH + 1
5810 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,LO
5815 C2 = PEEK (M + 3):
      C2 = PEEK (M + 3)
5820 LW = PEEK (M + 6):
      LW = PEEK (M + 6)
5825 VTAB 6: PRINT "UPSENSOR READING ="C2
5830 IF C2 < 131 THEN GOTO 5945

```



```

5835 IF LW > 160 THEN GOTO 5985
5840 DH = DH + 1
5845 POKE L1,HI:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
5850 IF C2 > C1 THEN Z = 0:G = 0
5855 IF C2 < C1 THEN Z = Z + 1:G = G + 2
5860 IF (C2 = C1) AND (Z > 0) THEN G = G + 2
5865 C1 = C2
5870 IF Z = 3 THEN AB = 3: GOTO 6090
5875 DH = DH + 1
5880 REM
5885 REM *** RETURN TO STEP ONE          ***
5890 REM
5895 GOTO 5270
5900 REM
5905 REM *** IF UPSENSOR GOES BELOW      ***
5910 REM *** THRESHOLD RETURN TO START ***
5915 REM *** POSITION. USE FLASHING      ***
5920 REM *** MESSAGE. REZERO COUNTER   ***
5925 REM *** (AB) SINCE SUN LOSS DUE   ***
5930 REM *** TO CLOUD COVER IN NORMAL  ***
5935 REM *** OCCURANCE.                ***
5940 REM
5945 VTAB 12: FLASH :
      PRINT "SUN INTENSITY BELOW THRESHOLD VALUE":
      VTAB 13: PRINT "RETURNING TO START UP POSITION":
      NORMAL :AB = 0: GOTO 1470
5950 REM
5955 REM *** IF TRANSITION FROM EAST TO***
5960 REM *** WEST WITHOUT ACHIEVING     ***
5965 REM *** LOCKON, RETURN TO STARTUP ***
5970 REM *** POSITION, BUT DO NOT RESET***
5975 REM *** COUNTER (AB).              ***
5980 REM
5985 VTAB 12: FLASH :
      PRINT "REACHED WEST LIMIT WITHOUT FINDING":
      VTAB 13: PRINT "ACCEPTABLE LOCK ON CRITERIA, RESET":
      NORMAL : GOTO 1470
5990 REM
5995 REM *** IF FAIL TO FIND SUN AFTER ***
6000 REM *** TWO EAST TO WEST TRYs,     ***
6005 REM *** PRINT ERROR MESSAGE AND    ***
6010 REM *** DETERMINE IF WANT TO      ***
6015 REM *** CONTINUE TRYING.          ***
6020 REM
6025 VTAB 16: FLASH :
      PRINT "UNABLE TO LOCATE SUN LONGITUDE"
6030 VTAB 17: PRINT "AFTER TWO TRYs. SUN PROBABLY"
6035 VTAB 18: PRINT "SETTING TOO FAR WEST": NORMAL

```



```

6040 VTAB 20: INPUT "WANT TO TRY AGAIN(Y/N)?";K$
6045 IF K$ = "Y" THEN AB = 0: HOME : GOTO 1460
6050 IF K$ = "N" THEN PRINT CHR$(4);"RUN HELLO"
6055 REM
6060 REM ***      BACKUP STEP SEQUENCE      ***
6065 REM
6070 REM *** RETURN TO POSITION WHERE ***
6075 REM *** SUN INTENSITY FIRST BEGAN ***
6080 REM *** TO DROP OFF (G) STEPS AGO ***
6085 REM
6090 POKE L1,HI:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
6095 GOSUB 6205
6100 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,LO
6105 GOSUB 6205
6110 POKE L1,HI:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
6115 GOSUB 6205
6120 POKE L1,LO:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
6125 GOSUB 6205
6130 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,HI
6135 GOSUB 6205
6140 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,LO
6145 GOSUB 6205
6150 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
6155 GOSUB 6205
6160 POKE L1,LO:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
6165 GOSUB 6205
6170 GOTO 6090

```

```

6175 REM
6180 REM *** SUBROUTINE TO DECRIMENT ***
6185 REM *** (G) COUNTER, AND DEGREES ***
6190 REM *** HORIZONTAL COUNT AS ***
6195 REM *** BACKUP. ***
6200 REM
6205 IF G = 0 THEN AC = 0: GOTO 6260
6210 G = G - 1
6215 DH = DH - 1
6220 RETURN.
6225 REM
6230 REM *** THIS PORTION OF THE SUN ***
6235 REM *** LOCATION SECTION GETS AN ***
6240 REM *** ELEVATION LOCKON PRIOR TO ***
6245 REM *** PASSING CONTROL TO THE ***
6250 REM *** TRACKING SECTION. ***
6255 REM
6260 DV = 0
6265 REM
6270 REM *** IF GO THROUGH ELEVATION ***
6275 REM *** TRANSITION TWICE WITHOUT ***
6280 REM *** ACHIEVING LOCKON, PRINT ***
6285 REM *** ERROR MESSAGE. ***
6290 REM
6295 IF AC = 2 THEN GOTO 5365
6300 AC = AC + 1
6305 HOME
6310 VTAB 2: PRINT "NOW LOCKING ONTO SUN'S ELEVATION"
6315 REM
6320 REM *** PRINT LONGITUDE VALUE ***
6325 REM *** FOUND IN PREVIOUS ROUTINE ***
6330 REM *** WITH ONE DECIMAL POINT ***
6335 REM *** ACCURACY. ***
6340 REM
6345 X1 = DH / LD
6350 X1 = INT (X1 * 10 + .5) / 10
6355 VTAB 4: PRINT "SUN LONGITUDE ="X1
6360 VTAB 5: PRINT "EAST=0 DEG SOUTH=90 DEG WEST =180 DEG"
6365 REM
6370 REM *** SET MEMORY LOCATIONS FOR ***
6375 REM *** DRIVE ROUTINE. ***
6380 REM
6385 L1 = M + 5:
        L2 = M + 6:
        L3 = M + 7:
        L4 = M + 8
6390 POKE L1,LO:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
6395 REM

```

```

6400 REM *** GET VALUES OF UPSENSOR (C3) ***
6405 REM *** DOWN SENSOR (C4), AND ***
6410 REM *** UP LIMIT SWITCH (LU) ***
6415 REM
6420 C3 = PEEK (M + 3):
      C3 = PEEK (M + 3)
6425 C4 = PEEK (M + 4):
      C4 = PEEK (M + 4)
6430 LU = PEEK (M + 7):
      LU = PEEK (M + 7)
6435 REM
6440 REM *** IF UPSENSOR (C3) BELOW ***
6445 REM *** SUNLIGHT THRESHOLD, OR HIT ***
6450 REM *** UPLIMIT SWITCH, EXIT DRIVE ***
6455 REM *** LOOP. ***
6460 REM
6465 IF C3 < 131 THEN GOTO 6920
6470 IF LU > 160 THEN GOTO 6950
6475 REM
6480 REM *** INCRIMENT DEGREES ELEVA- ***
6485 REM *** TION COUNTER (DV) ***
6490 REM
6495 DV = DV + 1
6500 REM
6505 REM *** STEP TWO ***
6510 REM
6515 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
6520 REM
6525 REM *** IF ACHIEVE BALANCE BETWEEN ***
6530 REM *** UPPER AND LOWER SENSOR TO ***
6535 REM *** WITHIN ONE UNIT, GO TO ***
6540 REM *** SUN TRACKING ROUTINE. ***
6545 REM
6550 IF C3 = C4 THEN GOTO 8060
6555 IF (C3 = C4 + 1) OR (C3 = C4 - 1) THEN GOTO 8060
6560 DV = DV + 1
6565 REM
6570 REM *** CALCULATE DEGREES ELEV. ***
6575 REM *** BY DIVIDING THE COUNTER ***
6580 REM *** VALUE (DV) BY STEPS PER ***
6585 REM *** DEGREE ELEVATION (LE), ***
6590 REM *** DETERMINED IN CALIBRATION ***
6595 REM *** ROUTINE. REPEAT ABOVE ***
6600 REM *** LOGIC FOR REMAINING STEPS ***
6605 REM
6610 X2 = DV / LE
6615 POKE L1,LO:
      POKE L2,HI:

```

```

        POKE L3,LO:
        POKE L4,LO
6620   C3 = PEEK (M + 3):
        C3 = PEEK (M + 3)
6625   C4 = PEEK (M + 4):
        C4 = PEEK (M + 4)
6630   LU = PEEK (M + 7):
        LU = PEEK (M + 7)
6635   IF C3 < 131 THEN GOTO 6920
6640   IF LU > 160 THEN GOTO 6950
6645   DV = DV + 1
6650   POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
6655   IF C3 = C4 THEN GOTO 8060
6660   IF (C3 = C4 + 1) OR (C3 = C4 - 1) THEN GOTO 8060
6665   DV = DV + 1
6670   X2 = INT (X2 * 10 + .5) / 10
6675   POKE L1,LO:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
6680   C3 = PEEK (M + 3):
        C3 = PEEK (M + 3)
6685   C4 = PEEK (M + 4):
        C4 = PEEK (M + 4)
6690   LU = PEEK (M + 7):
        LU = PEEK (M + 7)
6695   IF C3 < 131 THEN GOTO 6920
6700   IF LU > 160 THEN GOTO 6950
6705   DV = DV + 1
6710   POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
6715   IF C3 = C4 THEN GOTO 8060
6720   IF (C3 = C4 + 1) OR (C3 = C4 - 1) THEN GOTO 8060
6725   DV = DV + 1
6730   REM
6735   REM *** PRINT DEGREES ELEVATION ***
6740   REM *** ONLY ONCE PER EIGHT STEP ***
6745   REM *** LOOP DUE TO SPEED ***
6750   REM *** CONSIDERATIONS. ***
6755   REM
6760   VTAB 7: PRINT "DEGREES ELEVATION ="X2" DEG"
6765   POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
6770   C3 = PEEK (M + 3):

```

```

        C3 = PEEK (M + 3)
6775    C4 = PEEK (M + 4):
        C4 = PEEK (M + 4)
6780    LU = PEEK (M + 7):
        LU = PEEK (M + 7)
6785    IF C3 < 131 THEN GOTO 6920
6790    IF LU > 160 THEN GOTO 6950
6795    DV = DV + 1
6800    POKE L1,HI:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
6805    IF C3 = C4 THEN GOTO 8060
6810    IF (C3 = C4 + 1) OR (C3 = C4 - 1) THEN GOTO 8060
6815    DV = DV + 1
6820    GOTO 6390
6825    REM
6830    REM *** PRINT ERROR MESSAGE IF      ***
6835    REM *** HAVENT ACHIEVED LOCKON      ***
6840    REM *** AFTER TWO ATTEMPTS, AND      ***
6845    REM *** DETERMINE IF WANT TO        ***
6850    REM *** CONTINUE.                    ***
6855    REM
6860    VTAB 16: FLASH :
        PRINT "UNABLE TO ACHIEVE SENSOR BALANCE AFTER"
6865    VTAB 17:
        PRINT "TWO ATTEMPTS. POSSIBLE SYSTEM PROBLEM": NORMAL
6870    VTAB 19: INPUT "WANT TO TRY AGAIN(Y/N)?";K$
6875    IF K$ = "Y" THEN AC = 0: HOME : GOTO 1460
6880    IF K$ = "N" THEN PRINT CHR$(4);"RUN HELLO"
6885    REM
6890    REM *** IF SUN GOES BELOW THRES-    ***
6895    REM *** HOLD (BEHIND A CLOUD)      ***
6900    REM *** WHILE ATTEMPTING TO LOCK   ***
6905    REM *** ON, RETURN TO START POSIT.***
6910    REM *** AND TRY AGAIN.              ***
6915    REM
6920    VTAB 12: FLASH :
        PRINT "SUN INTENSITY BELOW THRESHOLD VALUE":
        VTAB 13: PRINT "RETURNING TO START UP POSITION":
        AC = 0: NORMAL : GOTO 1470
6925    REM
6930    REM *** IF REACH UPPER LIMIT        ***
6935    REM *** SWITCH AFTER FIRST TRY,     ***
6940    REM *** PRINT THIS MESSAGE.        ***
6945    REM
6950    VTAB 12: FLASH :
        PRINT "HAVE REACHED UPPER LIMIT WITHOUT":
        VTAB 13: PRINT "ACHIEVING SENSOR BALANCE, REZEROING":
        NORMAL : GOTO 1470

```



```

8000 REM
8010 REM *****
8020 REM **          SUN TRACKING ROUTINE          **
8030 REM *****
8040 REM
8050 REM
8060 HOME
8070 REM
8080 REM *** ONCE ROUGH LOCKON IS          ***
8090 REM *** ACHIEVED BY THE PREVIOUS ***
8100 REM *** ROUTINE, THIS ROUTINE FINE***
8110 REM *** ADJUSTS SENSOR BALANCE AND***
8120 REM *** TRACKS THE SUN.          ***
8130 REM
8140 REM *** INIT. OUT OF BALANCE CTRS.***
8150 REM
8160 UN = 0:
      DN = 0:
      EN = 0:
      WN = 0
8170 REM
8180 REM *** COUNTER USED TO DETERMINE ***
8190 REM *** IF SENSOR UNBALANCE IS ***
8200 REM *** VALID OR TRANSITORY ***
8210 REM
8220 Q = 5
8230 VTAB 2: PRINT "SUN TRACKING ROUTINE"
8240 VTAB 3: PRINT "THIS ROUTINE INSURES THAT THE TRACKER"
8250 VTAB 4: PRINT "STAYS LOCKED ONTO THE SUN. IF A CLOUD"
8260 VTAB 5: PRINT "COVERS THE SUN, CONTROL IS PASSED TO"
8270 VTAB 6: PRINT "THE CLOUD COVER WAIT ROUTINE UNTIL"
8280 VTAB 7: PRINT "THE SUN REAPPEARS"
8290 REM
8300 REM *** GET VALUES OF ALL SENSORS ***
8310 REM *** AND LIMIT SWITCHES. ***
8320 REM
8330 US = PEEK (M + 3):
      US = PEEK (M + 3)
8340 DS = PEEK (M + 4):
      DS = PEEK (M + 4)
8350 ES = PEEK (M + 1):
      ES = PEEK (M + 1)
8360 WS = PEEK (M + 2):
      WS = PEEK (M + 2)
8370 UL = PEEK (M + 7):
      UL = PEEK (M + 7)
8380 DL = PEEK (M + 8):
      DL = PEEK (M + 8)
8390 EL = PEEK (M + 5):
      EL = PEEK (M + 5)

```



```

8400  WL = PEEK (M + 6):
      WL = PEEK (M + 6)
8410  REM
8420  REM *** IF UUPSENSOR GOES BELOW ***
8430  REM *** THRESHOLD GO TO CLOUD ***
8440  REM *** WAIT ROUTINE. IF HIT LIMIT***
8450  REM *** SWITCH, GO TO APPROPRIATE ***
8460  REM *** NOTIFICATION MESSAGE ***
8470  REM
8480  IF US < 131 THEN GOTO 11090
8490  IF UL > 160 THEN GOTO 9220
8500  IF DL > 160 THEN GOTO 9270
8510  IF EL > 160 THEN GOTO 9320
8520  IF WL > 160 THEN GOTO 9370
8530  REM
8540  REM *** IF UPPER/LOWER OR EAST/ ***
8550  REM *** WEST SENSORS ARE IN ***
8560  REM *** BALANCE, ZERO APPROPRIATE ***
8570  REM *** UNBALANCE COUNTER. ***
8580  REM
8590  IF US = DS THEN UN = 0:DN = 0
8600  IF ES = WS THEN EN = 0:WN = 0
8610  REM
8620  REM *** IF SENSOR PAIRS ARE OUT OF***
8630  REM *** BALANCE INCRIMENT PROPER ***
8640  REM *** UNLBALANCE COUNTER. ***
8650  REM
8660  IF US > DS THEN UN = UN + 1
8670  IF DS > US THEN DN = DN + 1
8680  IF ES > WS THEN EN = EN + 1
8690  IF WS > ES THEN WN = WN + 1
8700  REM
8710  REM *** SYSTEM OPERATION OUTPUTS ***
8720  REM
8730  VTAB 17: PRINT "UN="UN" DN="DN" EN="EN" WN="WN"
      L5="L5"
8740  VTAB 18: PRINT "SU="SU" SD="SD" SE="SE" SW="SW"
8750  REM
8760  REM *** CHECK SENSOR OUT OF ***
8770  REM *** BALANCE COUNTERS AND GO TO***
8780  REM *** APPROPRIATE DRIVE ROUTINE ***
8790  REM *** IF COUNTER IS ABOVE ***
8800  REM *** THRESHOLD VALUE (G) ***
8810  REM
8820  IF UN > = Q THEN L1 = M + 5:
      L2 = M + 6:
      L3 = M + 7:
      L4 = M + 8:
      L5 = 1:
      GOTO 9480
8830  IF DN > = Q THEN L1 = M + 5:

```

```

      L2 = M + 6:
      L3 = M + 7:
      L4 = M + 8:
      L5 = 1:
      GOTO 10380
8840  IF EN > = Q THEN L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = 2:
      GOTO 10380
8850  IF WN > = Q THEN L1 = M + 1:
      L2 = M + 2:
      L3 = M + 3:
      L4 = M + 4:
      L5 = 2:
      GOTO 9480
8860  REM
8870  REM *** PUT SENSOR VALUES ON      ***
8880  REM *** SCREEN                      ***
8890  REM
8900  VTAB 9: PRINT TAB( 15)"US="US
8910  VTAB 10: PRINT TAB( 12)"WS="WS" ES="ES
8920  VTAB 11: PRINT TAB( 15)"DS="DS
8930  REM
8940  REM *** CONTINUE LOOP              ***
8950  REM
8960  GOTO 8330
8970  REM
8980  REM *** PRINT DEGREES LONGITUDE    ***
8990  REM *** AND ELEVATION VALUES WHEN ***
9000  REM *** RETURN FROM DRIVE ROUTINES***
9010  REM *** ONLY, SO LOGIC LOOP IS NOT***
9020  REM *** SLOWED DOWN BY UNNECESSARY***
9030  REM *** PRINT STATEMENTS.          ***
9040  REM
9050  X1 = DH / LD
9060  X1 = INT (X1 * 10 + .5) / 10
9070  VTAB 14: PRINT "DEGREES LONGITUDE ="X1" DEG"
9080  X2 = DV / LE
9090  X2 = INT (X2 * 10 + .5) / 10
9100  VTAB 15: PRINT "DEGREES ELEVATION ="X2" DEG"
9110  REM
9120  REM *** RETURN TO LOGIC LOOP AFTER***
9130  REM *** PRINT OUT.                  ***
9140  REM
9150  GOTO 8330
9160  REM
9170  REM *** LIMIT SWITCH DEPRESSION     ***
9180  REM *** MESSAGES PRINTED WHEN HIT  ***
9190  REM *** LIMIT SWITCH WHILE IN      ***

```

```

9200 REM *** TRACKING MODE. ***
9210 REM
9220 VTAB 17: FLASH : PRINT "HAVE REACHED UPPER LIMIT,
CHECK SYSYTEM"
9230 VTAB 18: PRINT "ALLIGNMENT TO TRUE EAST": NORMAL
9240 VTAB 19: INPUT "WANT TO RERUN(Y/N)?";K$
9250 IF K$ = "Y" THEN HOME : GOTO 1460
9260 IF K$ = "N" THEN PRINT CHR$ (4);"RUN HELLO"

9270 VTAB 17: FLASH : PRINT "HAVE REACHED DOWN LIMIT, MOST
PROBABLE"
9280 VTAB 18: PRINT "SUNSET OR MISALLIGNMENT": NORMAL
9290 VTAB 19: INPUT "WANT TO RERUN (Y/N)?";K$
9300 IF K$ = "Y" THEN HOME : GOTO 1460
9310 IF K$ = "N" THEN PRINT CHR$ (4);"RUN HELLO"

9320 VTAB 17: FLASH : PRINT "HAVE REACHED EAST LIMIT, MOST
PROBABLE"
9330 VTAB 18: PRINT "CAUSE EARLY MORNING, SUN TOO LOW":
NORMAL
9340 VTAB 19: INPUT "WANT TO RERUN (Y/N)?";K$
9350 IF K$ = "Y" THEN HOME : GOTO 1460
9360 IF K$ = "N" THEN PRINT CHR$ (4);"RUN HELLO"

9370 VTAB 17: FLASH : PRINT "HAVE REACHED WEST LIMIT, MOST
PROBABLE"
9380 VTAB 18: PRINT "CAUSE, NEAR SUNSET": NORMAL
9390 VTAB 19: INPUT "RERUN OR GO TO SLEEP (R/S)?";K$
9400 IF K$ = "R" THEN HOME : GOTO 1460
9410 IF K$ = "S" THEN HOME : VTAB 10:
PRINT "THANKYOU I'M REALLY TIRED": GOTO 1460
9420 REM
9430 REM *** UP/WEST DRIVE ROUTINES ***
9440 REM
9450 REM
9460 REM *** STEPS PER DRIVE ACCESS ***
9470 REM
9480 CT = 2
9490 REM
9500 REM *** GO TO DRIVE STEP ENTRY ***
9510 REM *** LOGIC STATEMENT AS DETER- ***
9520 REM *** MINED BY (L5) ***
9530 REM
9540 IF L5 = 1 THEN GOTO 9620
9550 IF L5 = 2 THEN GOTO 9630
9560 REM
9570 REM *** DETERMINE PROPER STEP TO ***
9580 REM *** REENTER DRIVE ROUTINE TO ***
9590 REM *** PREVENT JERKY SEEKER ***
9600 REM *** MOTION WHILE TRACKING ***
9610 REM

```

```

9620  ON SU GOTO 9640,9950,10000,10050,
      10100,10150,10200,10250
9630  ON SW GOTO 9640,9950,10000,10050,
      10100,10150,10200,10250
9640  POKE L1,L0:
      POKE L2,L0:
      POKE L3,HI:
      POKE L4,L0
9650  REM
9660  REM *** SET REENTRY STEP FOR THIS ***
9670  REM *** AND CORRESPONDING DRIVE ***
9680  REM *** LOOP IN OPPOSITE DIRECTION***
9690  REM *** SINCE A STEP HERE CHANGES ***
9700  REM *** THE REENTRY POINT IN OTHER***
9710  REM *** DRIVE LOOP. ***
9720  REM
9730  IF L5 = 1 THEN SU = 2:
      DV = DV + 1:UN = 0:SD = 1
9740  REM
9750  REM *** BESIDES DETERMINING ***
9760  REM *** REENTRY POINT, CHANGE ***
9770  REM *** DEGREE COUNTERS (DV),(DH) ***
9780  REM *** AND ZERO UNBALANCE COUNTER***
9790  REM *** (UN),(WN)
9800  REM
9810  IF L5 = 2 THEN SW = 2:
      DH = DH + 1:WN = 0:SE = 1
9820  REM
9830  REM *** AFTER CORRECT NUMBER OF ***
9840  REM *** STEPS PER DRIVER ACCESS ***
9850  REM *** RETURN TO LOGIC SECTION ***
9860  REM
9870  IF CT = 0 THEN GOTO 9050
9880  REM
9890  REM *** DECRIMENT STEPS PER ACCESS***
9900  REM *** COUNTER. CONTINUE ABOVE ***
9910  REM *** LOGIC SEQUENCE FOR EIGHT ***
9920  REM *** STEP LOOP. ***
9930  REM
9940  CT = CT - 1
9950  POKE L1,L0:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,L0
9960  IF L5 = 1 THEN SU = 3:
      DV = DV + 1:UN = 0:SD = 8
9970  IF L5 = 2 THEN SW = 3:
      DH = DH + 1:WN = 0:SE = 8
9980  IF CT = 0 THEN GOTO 9050
9990  CT = CT - 1
10000 POKE L1,L0:

```

```

        POKE L2,HI:
        POKE L3,LO:
        POKE L4,LO
10010  IF L5 = 1 THEN SU = 4:
        DV = DV + 1:UN = 0:SD = 7
10020  IF L5 = 2 THEN SW = 4:
        DH = DH + 1:WN = 0:SE = 7
10030  IF CT = 0 THEN GOTO 9050
10040  CT = CT - 1
10050  POKE L1,LO:
        POKE L2,HI:
        POKE L3,LO:
        POKE L4,HI
10060  IF L5 = 1 THEN SU = 5:
        DV = DV + 1:UN = 0:SD = 6
10070  IF L5 = 2 THEN SW = 5:
        DH = DH + 1:WN = 0:SE = 6
10080  IF CT = 0 THEN GOTO 9050
10090  CT = CT - 1
10100  POKE L1,LO:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
10110  IF L5 = 1 THEN SU = 6:
        DV = DV + 1:UN = 0:SD = 5
10120  IF L5 = 2 THEN SW = 6:
        DH = DH + 1:WN = 0:SE = 5
10130  IF CT = 0 THEN GOTO 9050
10140  CT = CT - 1
10150  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
10160  IF L5 = 1 THEN SU = 7:
        DV = DV + 1:UN = 0:SD = 4
10170  IF L5 = 2 THEN SW = 7:
        DH = DH + 1:WN = 0:SE = 4
10180  IF CT = 0 THEN GOTO 9050
10190  CT = CT - 1
10200  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
10210  IF L5 = 1 THEN SU = 8:
        DV = DV + 1:UN = 0:SD = 3
10220  IF L5 = 2 THEN SW = 8:
        DH = DH + 1:WN = 0:SE = 3
10230  IF CT = 0 THEN GOTO 9050
10240  CT = CT - 1
10250  POKE L1,HI:
        POKE L2,LO:

```



```

        POKE L3,HI:
        POKE L4,LO
10260  IF L5 = 1 THEN SU = 1:
        DV = DV + 1:UN = 0:SD = 2
10270  IF L5 = 2 THEN SW = 1:
        DH = DH + 1:WN = 0:SE = 2
10280  IF CT = 0 THEN GOTO 9050
10290  CT = CT - 1
10300  GOTO 9640
10310  REM
10320  REM *** DOWN/EAST DRIVE ROUTINES ***
10330  REM
10340  REM
10350  REM *** LOGIC SAME AS FOR UP/WEST ***
10360  REM *** DRIVE ROUTINES ***
10370  REM
10380  CT = 2
10390  IF L5 = 1 THEN GOTO 10410
10400  IF L5 = 2 THEN GOTO 10420
10410  ON SD GOTO 10430,10480,10530,10580,
        10630,10680,10730,10780
10420  ON SE GOTO 10430,10480,10530,10580,
        10630,10680,10730,10780
10430  POKE L1,HI:
        POKE L2,LO:
        POKE L3,HI:
        POKE L4,LO
10440  IF L5 = 1 THEN SD = 2:
        DV = DV - 1:DN = 0:SU = 1
10450  IF L5 = 2 THEN SE = 2:
        DH = DH - 1:EN = 0:SW = 1
10460  IF CT = 0 THEN GOTO 9050
10470  CT = CT - 1
10480  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,LO
10490  IF L5 = 1 THEN SD = 3:
        DV = DV - 1:DN = 0:SU = 8
10500  IF L5 = 2 THEN SE = 3:
        DH = DH - 1:EN = 0:SW = 8
10510  IF CT = 0 THEN GOTO 9050
10520  CT = CT - 1
10530  POKE L1,HI:
        POKE L2,LO:
        POKE L3,LO:
        POKE L4,HI
10540  IF L5 = 1 THEN SD = 4:
        DV = DV - 1:DN = 0:SU = 7
10550  IF L5 = 2 THEN SE = 4:
        DH = DH - 1:EN = 0:SW = 7

```



```

10560 IF CT = 0 THEN GOTO 9050
10570 CT = CT - 1
10580 POKE L1,LO:
      POKE L2,LO:
      POKE L3,LO:
      POKE L4,HI
10590 IF L5 = 1 THEN SD = 5:
      DV = DV - 1:DN = 0:SU = 6
10600 IF L5 = 2 THEN SE = 5:
      DH = DH - 1:EN = 0:SW = 6
10610 IF CT = 0 THEN GOTO 9050
10620 CT = CT - 1
10630 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,HI
10640 IF L5 = 1 THEN SD = 6:
      DV = DV - 1:DN = 0:SU = 5
10650 IF L5 = 2 THEN SE = 6:
      DH = DH - 1:EN = 0:SW = 5
10660 IF CT = 0 THEN GOTO 9050
10670 CT = CT - 1
10680 POKE L1,LO:
      POKE L2,HI:
      POKE L3,LO:
      POKE L4,LO
10690 IF L5 = 1 THEN SD = 7:
      DV = DV - 1:DN = 0:SU = 4
10700 IF L5 = 2 THEN SE = 7:
      DH = DH - 1:EN = 0:SW = 4
10710 IF CT = 0 THEN GOTO 9050
10720 CT = CT - 1
10730 POKE L1,LO:
      POKE L2,HI:
      POKE L3,HI:
      POKE L4,LO
10740 IF L5 = 1 THEN SD = 8:
      DV = DV - 1:DN = 0:SU = 3
10750 IF L5 = 2 THEN SE = 8:
      DH = DH - 1:EN = 0:SW = 3
10760 IF CT = 0 THEN GOTO 9050
10770 CT = CT - 1
10780 POKE L1,LO:
      POKE L2,LO:
      POKE L3,HI:
      POKE L4,LO
10790 IF L5 = 1 THEN SD = 1:
      DV = DV - 1:DN = 0:SU = 2
10800 IF L5 = 2 THEN SE = 1:
      DH = DH - 1:EN = 0:SW = 2
10810 IF CT = 0 THEN GOTO 9050

```

```

10820 CT = CT - 1
10830 GOTO 10430
11000 REM
11010 REM
11020 REM *****
11030 REM **          CLOUD COVER ROUTINE          **
11040 REM
11050 REM *** CLEAR SCREEN AND INITIAL- ***
11060 REM *** IZE LOOP COUNTERS FOR      ***
11070 REM *** TIME LOOPS.                ***
11080 REM
11090 HOME
11100 XX = 0:ZZ = 0
11110 VTAB 8: PRINT TAB( 5)"CLOUD COVER ROUTINE"
11120 VTAB 10: PRINT "THIS ROUTINE CHECKS THE UPPER SENSOR"
11130 VTAB 11: PRINT "WAITING FOR THE SUN TO COME BACK OUT"
11140 VTAB 12: PRINT "IF THE SUN HAS NOT COME BACK OUT FOR"
11150 VTAB 13: PRINT "APPROXIMATELY 10 MINUTES THE SYSTEM"
11160 VTAB 14: PRINT "WILL AUTOMATICALLY REZERO ITSELF AND"
11170 VTAB 15: PRINT "WAIT FOR THE SUNS INTENSITY TO AGAIN"
11180 VTAB 16: PRINT "GO ABOVE THE THRESHOLD VALUE"
11190 REM
11200 REM *** LOOPS TO ADJUST DELAY      ***
11210 REM *** TIMES FOR ONE SECOND CLOCK***
11220 REM
11230 FOR XX = 1 TO 605
11240 FOR ZZ = 1 TO 17
11250 REM
11260 REM *** CHECK UPPER SENSOR, AND    ***
11270 REM *** DISPLAY SENSOR AND TIME   ***
11280 REM *** DATA ON SCREEN.          ***
11290 REM
11300 US = PEEK (M + 3):
      US = PEEK (M + 3)
11310 VTAB 19: PRINT "UPPER SENSOR="US
11320 VTAB 20: PRINT "DELAY TIME COUNTER="XX" SEC"
11330 REM
11340 REM *** IF SUN COMES BACK OUT OR   ***
11350 REM *** REACH 10 MINUTE TIME OUT   ***
11360 REM *** EXIT CLOUD COVER ROUTINE  ***
11370 REM
11380 IF US > = 131 THEN GOTO 8060
11390 IF XX > = 600 THEN GOTO 1460
11400 NEXT ZZ
11410 NEXT XX

```

LIST OF REFERENCES

1. Howell, J. R., Bannerot, R. B., and Vilet, G. C., Solar-Thermal Energy Systems, McGraw-Hill Book Company, 1982.
2. Yurutucu, N., Microprocessor-Based Dual Axis Sun Tracker System, MSEE Thesis, Naval Postgraduate School, December 1982.
3. Selman, G. R., Straughen, A., Electric Machines, Addison-Wesley Publishing Company, 1980.
4. Mountain Computer Incorporated, A/D + D/A Card Operating Manual, Manual Number 11-00230-02, 1982.
5. California Computer Systems, CCS Model 7490 Apple II GPIB (IEEE) Interface Owner's Manual, Manual Number 89000-07490, 1981.
6. Apple Computer Incorporated, Applesoft II Basic Programming Reference Manual, Manual Number A2L0006, 1981.

INITIAL DISTRIBUTION LIST

| | No. Copies |
|--|------------|
| 1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314 | 2 |
| 2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943 | 2 |
| 3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943 | 1 |
| 4. Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943 | 5 |
| 5. Professor A. Gerba, Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943 | 1 |
| 6. Lieutenant Roger J. Morais, USN 10831 New Salem Point San Diego, California 92126 | 2 |
| 7. Captain J. E. Bley, Jr., Code 04 Director of Military Operations Naval Postgraduate School Monterey, California 93943 | 1 |

210457

Thesis

M81837 Morais

c.1

Integration of an
apple II plus computer
into an existing dual
axis sun tracker sys-
tem.

210457

Thesis

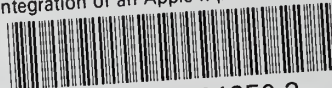
M81837 Morais

c.1

Integration of an
apple II plus computer
into an existing dual
axis sun tracker sys-
tem.

thesM81837

Integration of an Apple II plus computer



3 2768 001 01250 3

DUDLEY KNOX LIBRARY